



João Ivan Martins Aguiar Rodrigues e Silva

Licenciado em Engenharia Informática

Avaliação de Linguagens de Domínio Específico para Sistemas Multi-Agente

Dissertação para obtenção do Grau de
Mestre em Engenharia Informática

Orientador: Prof. Dr. Vasco Miguel Moreira do Amaral,
Prof. Auxiliar, Universidade Nova de Lisboa

Co-orientador: Prof. Dr. Miguel Carlos Pacheco Afonso Goulão,
Prof. Auxiliar, Universidade Nova de Lisboa

Júri:

Presidente: Prof. Dr. Matthias Knorr

Arguente: Prof. Dr. Jacomé Miguel Costa da Cunha

Vogal: Prof. Dr. Vasco Miguel Moreira do Amaral



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2018

Avaliação de Linguagens de Domínio Específico para Sistemas Multi-Agente

Copyright © João Ivan Martins Aguiar Rodrigues e Silva, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

AGRADECIMENTOS

Em primeiro lugar gostaria de agradecer aos meus orientadores, Professor Dr. Vasco Miguel Moreira do Amaral e Professor Dr. Miguel Carlos Pacheco Afonso Goulão, por toda a ajuda e orientação. Agradeço a disponibilidade, o apoio e dedicação, assim como as diversas indicações dadas ao longo da realização desta dissertação. Desenvolver esta dissertação sobre a vossa orientação foi uma experiência bastante motivadora e enriquecedora, tanto a nível profissional como pessoal.

Muito obrigada à Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa pela formação e condições oferecidas, assim como ao Departamento de Informática por proporcionar os recursos necessários para a realização dos estudos empíricos desta dissertação.

Ao meu colega Tomás Rogério Brochado de Miranda por toda a ajuda e disponibilidade.

A todos os voluntários que se disponibilizaram a participar nos estudos empíricos necessários nesta dissertação.

A todos os meus colegas e amigos, que me apoiaram, motivaram e estiveram sempre presentes em todos os momentos importantes do meu percurso.

À minha namorada pelo amor, compreensão e disponibilidade durante estes meses de trabalho.

E em último, mas com um agradecimento muito especial, à minha família, principalmente aos meus pais e irmãos. Muito obrigada pelo constante carinho, compreensão e pelo esforço que sempre dedicaram em prol do meu sucesso.

RESUMO

Sistemas Multi-Agente (SMA) formam uma sub-área da Inteligência Artificial Distribuída e concentram-se no estudo de agentes com o intuito de resolver os seus problemas para atingir determinados objetivos.

A *Semantic Web Enabled Agent Modeling Language* (SEA_ML) é uma *Linguagem de Domínio Específico* (LDE) usada até ao momento para fins educacionais que permite que sejam criados SMA num alto nível de abstração.

A referida LDE foi criada com a necessidade de solucionar problemas relacionados com a produtividade dos *developers*. Embora tendo expressividade reduzida em termos computacionais, uma vez que pretende ser mais abstrata, permite manipular conceitos do domínio. Esta é construída com o propósito de ser mais próximo da forma de pensar dos utilizadores de um determinado domínio. Para que uma LDE contribua para a produtividade dos seus utilizadores, tem de garantir usabilidade, sendo naturalmente necessário que cumpra com requisitos de usabilidade. A usabilidade está relacionada com a *Interação Pessoa Máquina* (IPM), estudo sobre a forma como a tecnologia influencia as atividades humanas, cujo objetivo é que os utilizadores possam alcançar os seus objetivos com eficácia, eficiência e satisfação.

Apesar de já ter sido feita uma avaliação e consequentemente uma melhoria nas notações do SEA_ML, originando o SEA_ML++, verifica-se que os utilizadores desta continuam a apresentar um elevado grau de dificuldade na aprendizagem e uso desta linguagem, apontando a falta de usabilidade como o principal problema.

Com vista a identificar, e se possível corrigir, o problema de usabilidade, foi realizado trabalho apresentado nesta dissertação que inclui uma segunda iteração à avaliação da usabilidade da linguagem SEA_ML, sendo que foram realizadas duas experiências. A primeira consistiu no estudo sobre a avaliação da transparência semântica, tendo sido identificadas oportunidades de melhoria para a sintaxe concreta desta linguagem através de um conjunto de questionários.

O outro estudo em que se pretende comparar o SEA_ML++ com uma outra LDE, pretendeu-se avaliar a forma como os participantes interagem com ambas as linguagens, sendo que posteriormente foram analisadas algumas métricas como a taxa de sucesso, esforço e dados de *eye-tracking* recolhidos durante o processo experimental.

Os resultados obtidos indicam que os participantes apresentam uma maior taxa de

satisfação ao desenvolver um SMA com o SEA_ML++. Contudo, existe um maior esforço cognitivo e uma maior complexidade no uso desta linguagem.

No final, foram propostas algumas melhorias na linguagem SEA_ML++, de modo a fortalecer a linguagem no ponto de vista da sua usabilidade.

Palavras-chave: Linguagem de Domínio Específico, Interação Pessoa-Máquina, Semantic Web Enabled Agent Modeling Language, Engenharia de Software Experimental

ABSTRACT

Multi-Agent Systems (SMA) form a sub-area of Distributed Artificial Intelligence and focus on the study of agents in order to solve their problems to achieve certain objectives.

The Semantic Web Enabled Agent Modeling Language (SEA_ML) is a Domain-Specific Language (LDE) used so far for educational purposes that allows SMA to be created at a high level of abstraction.

This LDE was created with the need to solve problems related to the productivity of the developers. Although having reduced expressiveness in computational terms, since it is intended to be more abstract, it allows manipulating domain concepts. This is built with the purpose of being closer to the way of thinking of the users of a particular domain. For an LDE to contribute to the productivity of its users, it has to guarantee usability, and of course it must comply with usability requirements. Usability is related to Person Machine Interaction (IPM), a study on how technology influences human activities, whose goal is for users to achieve their goals with efficiency, effectiveness and satisfaction.

Although an evaluation has been made and consequently an improvement in the SEA_ML notations, it is verified that the users of this one continue to present a high degree of difficulty in the learning and use of this language, pointing the lack of usability as the main problem.

In order to identify, and if possible to correct, the usability problem, a work presented in this dissertation was included, which includes a second iteration to the evaluation of the usability of the SEA_ML language. Two experiments were carried out. The first one consisted of the study on the evaluation of semantic transparency, and opportunities for improvement were identified for the concrete syntax of this language through a set of questionnaires.

The other study in which SEA_ML++ is intended to be compared with another LDE was to evaluate the way in which the participants interact with both languages. Some metrics such as the success rate, effort and eye-tracking data collected during the experimental process.

The results indicate that the participants present a higher satisfaction rate when developing an SMA with SEA_ML++. However, there is a greater cognitive effort and a greater complexity in the use of this language.

In the end, some improvements were proposed in the SEA_ML++ language, in order to strengthen the language in the point of view of its usability.

Keywords: Domain Specific Languages, Human-Computer Interaction, Semantic Web Enabled Agent Modeling Language, Experimental Software Engineering

CONTEÚDO

Lista de Figuras	xv
Lista de Tabelas	xvii
1 Introdução	1
1.1 Contexto	1
1.2 Motivação	2
1.3 Objetivos	3
1.4 Principais Contribuições	3
1.5 Estrutura do Documento	4
2 Conceitos Gerais	5
2.1 Linguagens de Domínio Específico	5
2.1.1 Vantagens e Desvantagens	6
2.1.2 Ciclo de Desenvolvimento	6
2.1.3 Stakeholders	7
2.1.4 Paradigmas de Desenvolvimento	8
2.1.5 Ferramentas de Desenvolvimento	8
2.1.6 Plataformas Alvo	9
2.2 Usabilidade	9
2.3 Notações Visuais	10
2.3.1 Physics Of Notations	11
2.4 Eye Tracker	12
2.4.1 Métricas	13
2.5 System Usability Scale	14
2.5.1 Pontuação SUS	14
2.6 Web Semântica	15
2.7 Sistemas Multi-Agente	15
2.7.1 Linguagens para SMA	16
3 Ambientes de Desenvolvimento para Sistemas Multi-Agente	19
3.1 SEA_ML	20
3.1.1 Sintaxe Abstrata	20

3.1.2	Sintaxe Concreta	21
3.1.3	Exemplo: Troca de Carros	21
3.2	DSML4MAS	24
3.2.1	Sintaxe Abstrata	24
3.2.2	Sintaxe Concreta	24
3.2.3	Exemplo: Troca de Carros	25
4	Trabalhos Relacionados	27
4.1	Abordagem para Desenhar e Avaliar a Eficácia Cognitiva de Sintaxes Concretas	27
4.1.1	Processo de Desenvolvimento do Estudo	28
4.1.2	Análise dos Resultados	29
4.2	Abordagem Sistemática para avaliar ambientes de linguagem de domínio específico para sistemas multi-agente	29
4.2.1	Processo de Desenvolvimento do Estudo	29
4.2.2	Análise dos Resultados	30
4.3	Avaliar a Framework SEA_ML	31
4.3.1	Processo de Desenvolvimento do Estudo	31
4.3.2	Análise dos Resultados	31
4.4	Análise Comparativa	32
5	Preparação da Avaliação Experimental	33
5.1	Análise dos Objetivos	33
5.2	Processo Experimental	35
6	Avaliação Transparência Semântica SEA_ML e SEA_ML++	37
6.1	Objetivos	37
6.2	Material Experimental	38
6.3	Participantes	38
6.4	Plano de Execução	39
6.5	Análise dos Resultados	39
6.6	Teste de Hipótese	46
6.7	Ameaças à validade do estudo	47
7	Avaliar a SEA_ML++ comparando-a com o Domain Specific Modeling Language for Multi-Agent Systems (DSML4MAS)	49
7.1	Objetivos	49
7.2	Tarefas	49
7.3	Material Experimental	51
7.4	Participantes	51
7.5	Caso de estudo	51
7.5.1	Reserva de uma Viagem Empresarial	52

7.5.2	Reposição de Armazém	56
7.6	Plano de Execução	60
7.7	Preparação dos Dados	61
7.7.1	Ogama	61
7.7.2	Heatmap	62
7.8	Análise dos Dados	64
7.8.1	Estatística Descritiva	73
7.9	Ameaças à validade do estudo	78
8	Melhorias Propostas	79
8.0.1	Melhorias SEA_ML++	79
8.0.2	Melhorias DSML4MAS	81
9	Conclusões	83
9.1	Contribuições	84
9.2	Trabalho Futuro	84
	Bibliografia	85
A	Questionário Demográfico	91
B	Carta de Consentimento	93
C	Questionário de Satisfação	95
D	Exercício 1	99
E	Exercício 2	101

LISTA DE FIGURAS

2.1	Ciclo de desenvolvimento de uma LDE. Adaptado de [Bar+18].	7
2.2	Os 9 princípios de "Physics Of Notations". Adaptado de [Moo09].	11
2.3	Eye Tracker	12
2.4	Exemplo de um Caminho de Inspeção. Adaptado de [Yus+07].	13
2.5	Respostas possíveis do utilizados ao questionário System Usability Scale (SUS).	14
3.1	Ponto de Vista <i>Agent</i> para o exemplo Troca de Carros.	22
3.2	Ponto de Vista <i>MAS</i> para o exemplo Troca de Carros.	23
3.3	Ponto de Vista <i>SWS-Interaction</i> para o exemplo Troca de Carros.	23
3.4	Ponto de Vista <i>Agent</i> para o exemplo Troca de Carros.	25
3.5	Ponto de Vista <i>MAS</i> para o exemplo Troca de Carros.	26
3.6	Ponto de Vista <i>Organization</i> para o exemplo Troca de Carros.	26
5.1	Processo de Avaliação Usabilidade.	36
6.1	Plano de Execução	39
6.2	Porcentagem Respostas Corretas dos Participantes	40
6.3	Análise da Taxa de Sucesso (Verde = Acima do Limite; Sublinhado = Melhor)	41
6.4	Resultados sobre a Taxa de Sucesso	43
6.5	Coefficiente de Transparência Semântica	44
6.6	Resultados sobre o Coeficiente de Transparência Semântica	45
7.1	Tela Monitor	50
7.2	Desenho Experimental e Sequência de Tarefas	50
7.3	Resultados Questionário Demográfico	52
7.4	SEA_ML++ ponto de vista <i>Agent</i>	53
7.5	DSML4MAS ponto de vista <i>Agent</i>	53
7.6	SEA_ML++ ponto de vista <i>MasAndOrg</i>	54
7.7	DSML4MAS ponto de vista <i>Organization</i>	55
7.8	SEA_ML++ ponto de vista <i>SWSInteraction</i>	55
7.9	DSML4MAS ponto de vista <i>Mas</i>	56
7.10	SEA_ML++ ponto de vista <i>Agent</i>	57
7.11	DSML4MAS ponto de vista <i>Agent</i>	57
7.12	SEA_ML++ ponto de vista <i>MasandOrg</i>	58

7.13 DSML4MAS ponto de vista <i>Organization</i>	58
7.14 SEA_ML++ ponto de vista <i>SWSInteraction</i>	59
7.15 DSML4MAS ponto de vista <i>Mas</i>	60
7.16 Plano de Execução	61
7.17 Exemplo Heatmap sem Imagem de Fundo	62
7.18 Exemplo de Imagem de Fundo	63
7.19 Exemplo Heatmap	63
7.20 Métricas Eficácia Cognitiva	65
7.21 Ligação <i>Cooperates</i>	65
7.22 Agentes dentro das Organizações	66
7.23 Organizações e Ambientes	67
7.24 Áreas de Interesse	68
7.25 Métricas Eye Tracker	69
7.26 HeatMap Areas of Interest (AOI)- <i>Symbols</i>	70
7.27 HeatMaps	71
7.28 Elementos e Ligações	72
7.29 Recall para ambas as linguagens	74
7.30 Componentes de Qualidade de Nielsen	76
7.31 Métricas Precision e F-Measure	78
8.1 Seta Bidirecional entre dois Agentes	79
8.2 Agentes no <i>Compartment</i> da Organização	80
8.3 <i>Compartment Agent</i>	80
8.4 <i>Deployment Viewpoint</i>	81
8.5 Ligação entre Agentes e Organização	81
8.6 Ligação entre Organizações e <i>Environment</i>	81
8.7 <i>Compartment</i> para Organizações	82
8.8 DSML4MAS <i>Agent Viewpoint</i>	82

LISTA DE TABELAS

3.1	Sintaxe Concreta SEA_ML.	22
3.2	Sintaxe Concreta DSML4MAS.	25
6.1	Wilcoxon Test para a Taxa de Sucesso (a-SEA_ML++<SEA_ML, b-SEA_ML++>SEA_ML, c-SEA_ML++=SEA_ML)	46
6.2	Wilcoxon Test para o Coeficiente de Transparência Semântica (a-SEA_ML++<SEA_ML, b-SEA_ML++>SEA_ML, c-SEA_ML++=SEA_ML)	47
7.1	Estatística Descritiva	73
7.2	Welch's Test	74
7.3	Welch's Test	75
7.4	Welchs Test	76
7.5	Métrica Cliques e Duração para Participantes Principiantes	77
7.6	Welch's Test	78

GLOSSÁRIO

AOI Areas of Interest.

BDI Belief–Desire–Intention.

DSML4MAS Domain Specific Modeling Language for Multi-Agent Systems.

EMF Eclipse Modeling Framework.

GQM Goal Question Metric.

IPM Interação Pessoa Máquina.

ISO Organização Internacional de Padronização.

LDE Linguagem de Domínio Específico.

LPG Linguagem de Propósito Geral.

MDD Model Driven Development.

SEA_ML Semantic Web Enabled Agent Modeling Language.

SMA Sistemas Multi-Agente.

stakeholder Pessoa ou conjunto de pessoas que têm um determinado interesse num negócio. .

SUS System Usability Scale.

SWA Semantic Web Agents.

SWO Semantic Web Organization.

SWS Semantic Web Services.

TAO Taming Agents and Objects.

INTRODUÇÃO

1.1 Contexto

Uma [LDE](#), é uma linguagem de programação ou modelação criada para capturar expressões em termos do domínio do problema e não no domínio da solução. É útil por diversas razões, mas essencialmente por melhorar a produtividade dos utilizadores e permitir uma melhor comunicação com os [stakeholders](#) [[Fow10](#)].

[SMA](#) formam uma sub-área da Inteligência Artificial Distribuída e concentram-se no estudo de agentes autónomos num sistema multi-agente [[Fer99](#)]. Cada agente possui um conjunto de capacidades comportamentais que definem a autonomia necessária para utilizar as suas capacidades a fim de alcançar um determinado objetivo.

[SEA_ML](#) é uma linguagem usada até ao momento para fins educacionais, que possui um conjunto de ferramentas gráficas para modelar artefactos para [SMA](#), e como tal, permite que sejam criados sistemas de agentes num alto nível de abstração [[Cha+14](#)].

[DSML4MAS](#), é uma linguagem de modelação para domínio específico de [SMA](#) e permite também modelar sistemas de agentes de forma independente e com recurso a uma interface gráfica [[Hah08](#)].

A [IPM](#), é uma disciplina que tem crescido ao longo dos anos à medida que a indústria percebeu a importância da usabilidade nos seus produtos de software [[Dix09](#)]. Esta dissertação inclui várias técnicas de [IPM](#), como o *Eye Tracking*, uma técnica que permite verificar tanto para onde a pessoa está a olhar num determinado momento, como também verificar a sequência pela qual a pessoa percorreu uma determinada interface.

Esta dissertação faz parte de uma cooperação entre o centro de pesquisa NOVA LINC'S da Universidade Nova de Lisboa e o Instituto Internacional de Informática da *Ege Univeristy* (grupo que desenvolveu a linguagem em questão), na Turquia, que se refere ao projeto de desenvolvimento de uma estrutura de avaliação de [LDE](#) para [SMA](#).

1.2 Motivação

O aumento de novos domínios de aplicações e de recursos para computadores, levou à criação de novas linguagens de programação, nomeada das LDE's. São vários os relatos de benefícios de produtividade das LDEs, por exemplo, numa entrevista efetuada à *Defense Company* [Hut+14], fornecedora multinacional de equipamentos e serviços para as Forças Armadas, é afirmado que com o uso deste tipo de linguagens viram a sua produtividade a aumentar sete ou oito vezes mais. No entanto, a grande maioria são análises pouco sistemáticas e com pouca evidência científica, sendo que cada vez mais a comunidade se preocupa em estabelecer estudos empíricos válidos como sustento a estas afirmações. Para que uma linguagem seja usada corretamente e que seja possível atingir os objetivos esperados, é necessário que a LDE satisfaça um conjunto de requisitos de usabilidade. A usabilidade desempenha um papel importante na comunicação/expressão de um produto de software. De acordo com a *Organização Internacional de Padronização (ISO)* (do inglês *International Standards Organization*) 9241-11, é definida como a capacidade que um determinado produto de software tem de possibilitar aos utilizadores atingir os seus objetivos com eficácia, eficiência e satisfação [Bev+15]. Caso contrário, se o sistema não cumprir todos os requisitos de usabilidade, pode produzir resultados diferentes dos esperados.

Para garantir que uma LDE cumpre todos estes requisitos é necessário proceder a uma avaliação experimental propriamente desenvolvida. Numa primeira iteração efetuada à avaliação da linguagem *SEA_ML* [Mir17], foi avaliada a sintaxe concreta da linguagem com base na "*Physics Of Notations*", um conjunto de princípios para apoiar a avaliação de notações/símbolos das linguagens. Nesta avaliação, apesar de os participantes selecionarem corretamente um maior número de notações com a notação proposta em comparação com a notação original, foi possível concluir que os resultados não eram significativamente melhores para que tivesse um impacto na prática, em termos de tempo para resolver os problemas e precisão com que esses problemas eram resolvidos. Assim, especula-se que o principal problema para a falta de usabilidade do *SEA_ML*, não esteja relacionado com a sintaxe concreta, mas com o nível de abstração em que a linguagem se encontra estruturada.

Tendo em conta as razões anunciadas anteriormente, nesta dissertação foi necessário efetuar uma nova iteração à avaliação do *SEA_ML*. Assim, para além de se avaliar as notações do *SEA_ML++* que ficaram por avaliar na primeira avaliação feita à linguagem [Mir17], pretende-se que seja feita uma comparação do *SEA_ML++* com outra linguagem que permita modelar o mesmo sistema, *DSML4MAS*, de forma a ser possível detetar problemas de usabilidade, em relação à forma como a linguagem está desenhada, que possam estar presentes no *SEA_ML++* e que não estejam presentes no *DSML4MAS*.

1.3 Objetivos

Visto que na primeira iteração efetuada à avaliação da usabilidade do [SEA_ML](#), os resultados não foram os esperados [Mir17], esta dissertação tem como principal objetivo melhorar a sua usabilidade, e para tal identificar os problema atuais fazendo uma nova iteração à avaliação da usabilidade do [SEA_ML](#), desta vez envolvendo novas metodologias. Esta segunda iteração foi dividida em duas fases:

1. Avaliar um fator que ficou por avaliar na primeira iteração, a transparência semântica, que defende que os símbolos de uma linguagem devem sugerir o seu significado podendo assim ser facilmente deduzidos [Gur99]. O principal objetivo desta avaliação foi completar o trabalho que havia sido feito na primeira iteração [Mir17] (ver capítulo 6).
2. Avaliar a [SEA_ML++](#) comparando-a com outra linguagem que permita modelar o mesmo sistema, de forma a detetar aspetos que possam contribuir para melhorar o [SEA_ML++](#), como por exemplo o nível de abstração, a forma como se encontra desenhada a relação entre os vários conceitos ou até mesmo os padrões de desenho da linguagem. Para ser feita esta comparação, foram realizados um conjunto de testes com utilizadores, de modo a verificar quais as dificuldades que foram sentidas pelos utilizadores quando utilizam o [SEA_ML++](#), que não sentiram quando utilizaram o [DSML4MAS](#). Para a análise dos dados foram aplicadas métricas de forma a obter dados quantitativos e qualitativos que satisfaçam o nosso objetivo principal. Para este conjunto de métricas, tivemos a ajuda da tecnologia *Eye Tracking*, que permite definir o comportamento visual das pessoas (ver capítulo 7).

Depois de realizadas estas duas avaliações e de identificadas algumas falhas comuns por partes dos participantes no [SEA_ML++](#), foram também propostas modificações para ajudar a melhorar a usabilidade da linguagem.

1.4 Principais Contribuições

A principal contribuição desta dissertação foi a proposta de algumas melhorias à linguagem [SEA_ML++](#), melhorias essas que se encontram relacionadas com o nível de abstração que a linguagem se encontra estruturada, e a realização de dois estudos empíricos, envolvendo um total de 24 participantes.

Uma outra contribuição desta dissertação, foi a criação de um processo experimental que poderá ser usada em futuras investigações para a avaliação da usabilidade neste tipo de linguagens.

Este estudo foi publicado na *Third International Workshop on Human Factors in Modeling (HuFaMo'18)* [JSK18], um workshop de pesquisas empíricas em que o objetivo é melhorar o estado da ciência em assuntos relacionados com a Engenharia Baseada em Modelos.

1.5 Estrutura do Documento

Este documento está estruturado da seguinte forma:

1. **Capítulo 2 - Conceitos Gerais** - São apresentados conceitos relacionados com a Engenharia de Software, que são necessários para entender o trabalho proposto nesta dissertação. No final deste capítulo, será também apresentada a ferramenta [SEA_ML](#) e a [DSML4MAS](#).
2. **Capítulo 3 - Ambientes de Desenvolvimento para Sistemas Multi-Agente** - São apresentadas algumas [LDE's](#) para [SMA](#), nomeadamente a [SEA_ML](#) e o [DSML4MAS](#).
3. **Capítulo 4 - Trabalhos Relacionados** - São apresentados alguns trabalhos que estejam, de alguma forma, relacionados com esta dissertação. No fim, será feita uma análise comparativa entre as avaliações feitas à linguagem [SEA_ML](#) e à avaliação proposta.
4. **Capítulo 5 - Preparação da Avaliação Experimental** - Neste capítulo será apresentada uma proposta detalhada da avaliação a efetuar.
5. **Capítulo 6 - Avaliação Transparência Semântica [SEA_ML](#) e [SEA_ML++](#)** - Nesta capítulo é apresentado a primeira avaliação efetuada à linguagem [SEA_ML++](#).
6. **Capítulo 7 - Avaliar a [SEA_ML++](#) comparando-a com o [DSML4MAS](#)** - Nesta capítulo é apresentado a segunda avaliação efetuada à linguagem [SEA_ML++](#).
7. **Capítulo 8 - Melhorias Propostas** - Nesta capítulo são apresentadas algumas melhorias à linguagem [SEA_ML++](#).
8. **Capítulo 9 - Conclusões** - É apresentada uma síntese do trabalho realizado, onde se conclui e discute as futuras investigações que deverão ser tidos em conta para uma avaliação a esta linguagem.

CONCEITOS GERAIS

2.1 Linguagens de Domínio Específico

LDEs têm como objetivo acelerar a produtividade permitindo expressar conceitos ao nível de abstração do domínio do problema de uma forma rápida e completa de modo a que sejam possíveis de analisar pelos **stakeholders**, podendo assim verificar se o sistema cumpre todos os requisitos, reduzindo tempo de desenvolvimento [Fow10].

Sendo estas linguagens utilizadas para abstrair os detalhes do domínio da solução tecnológica, evitam que o utilizador tenha de se preocupar com questões como que tipo de estrutura a utilizar ou até mesmo que algoritmo implementar. Assim, para além de ser possível construir soluções rápidas, permitindo um ciclo de entrega mais curto e ágil, permite aos utilizadores do domínio gastar menos tempo em entender a implementação e a detetar problemas de linguagem [Cao+09].

As **LDEs** podem ser implementadas usando uma metodologia de Desenvolvimento Orientado a Modelos (do inglês **Model Driven Development (MDD)**), que inclusivamente promove que se crie abstrações sobre o domínio do problema como forma de lidar com a complexidade accidental [Bar+18].

Uma **LDE** é definida através dos seguintes componentes [SP11]:

- **Sintaxe Abstrata** - Representam os conceitos e as suas relações sem ter em consideração o seu significado. Em termos de **MDD**, a sintaxe abstrata é descrita através de um meta-modelo que define os conceitos que podem ser usados para criar um modelo que seja instância desse meta-modelo [Völ+13]
- **Sintaxe Concreta** - Fornece uma relação entre os elementos e as suas representações gráficas ou textuais. Assim sendo, a sintaxe concreta é um conjunto de notações que facilitam a apresentação e a construção da linguagem.

- **Semântica** - Define o significado dos programas de uma linguagem de programação. Tipicamente é feita de um modo informal através da implementação da tradução para o código correspondente ou modelos noutras linguagens.

Este tipo de linguagem pode ser aplicada em inúmeras áreas de domínio específico, como por exemplo, Excel (*spreadsheets*), SQL (*database queries*), LaTeX (*typesetting*), HTML (*hypertext web pages*) e MATLAB [Mer+05a].

2.1.1 Vantagens e Desvantagens

Ao comparar uma LDE com uma **Linguagem de Propósito Geral (LPG)**, em que as LDEs capturam abstrações, estas apresentam vantagens substanciais de expressividade e facilidade em termos de conceitos do domínio do problema, quando comparadas para um determinado domínio, traduzindo-se em proventos quer ao nível da produtividade e qualidade, quer ao nível de redução de custos de manutenção [Mer+05b]. Para além disso, o uso das LDEs pode servir como meio de comunicação visto ser possível fornecer aos **stakeholders** uma melhor compreensão do trabalho a ser desenvolvido, pois mesmo que o especialista do domínio não possa desenvolver, a sintaxe usada está próxima das notações usadas pelos especialistas do domínio [VD+00].

Por outro lado, as LDEs apresentam algumas desvantagens como o custo de desenhar, implementar e manter. Estes fatores devem ser tomado em conta na análise do custo-benefício. Por outro lado, seria também necessário investir na formação dos utilizadores que vão utilizar a LDE, o que traduz num elevado custo [VD+00]. Para além disso, são mais difíceis de modificar e integrar com outros componentes do sistema, pois as arquiteturas de hoje em dia ainda não se encontram formalizadas para integrar uma LDE no processo de Software [Spi01]. Sendo também uma tecnologia que se encontra em fase de expansão, a falta de suporte de ferramentas para o desenvolvimento e manutenção de LDEs é ainda um problema.

2.1.2 Ciclo de Desenvolvimento

Para criar uma LDE, devem ser seguidas algumas etapas de forma a obter o cenário que o cliente pretende. De acordo com [Bar+18], para o desenvolvimento de uma LDE é necessário um processo iterativo que envolva as seguintes fases:

1. **Decisão** - Devem reunir-se todos os **stakeholders**, de forma a identificar a necessidade de uma LDE para o domínio do problema e a sua validade, tendo em conta o investimento necessário, não só no processo de desenvolvimento, mas também na sua manutenção.
2. **Análise do Domínio** - Nesta fase é identificado o domínio do problema, onde os clientes e os Engenheiros são participantes ativos no projeto. O resultado final desta fase, consiste num modelo de domínio que contém uma definição do domínio

que se pretende atingir, a nomenclatura utilizada, uma descrição dos conceitos do domínio e por último um Modelo de Características (do inglês *Feature Model*). Um dos resultados mais importante da Análise do Domínio é o Modelo de Características, sendo considerado um elemento chave pois através de uma representação gráfica, é possível obter uma melhor compreensão de todas as características necessárias para o desenvolvimento da LDE [VDK02].

3. Desenho - Criar o meta-modelo da LDE, de forma a permitir que os utilizadores visualizem e utilizem o meta-modelo para criar os seus próprios modelos.
4. Implementação - Escolher e implementar a abordagem mais adequada para a linguagem, com base na análise do domínio e no meta-modelo criado.
5. Validação - Verificar e validar a linguagem, através de testes para garantir a sua integridade.
6. Implantação - Se todas as etapas anteriores foram concluídas com sucesso, então a LDE está pronta a ser usada. Especialistas do domínio e utilizadores do domínio devem especificar modelos com a LDE criada, respeitando os padrões mencionados nas duas primeiras fases de desenvolvimento.

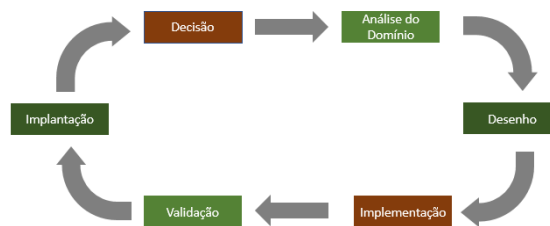


Figura 2.1: Ciclo de desenvolvimento de uma LDE. Adaptado de [Bar+18].

Ao utilizar este processo, caso no fim de todas as fases do processo a linguagem não cumpra todos os requisitos, é possível realizar novas iterações para que a linguagem possa ir evoluindo no seu ciclo de vida. A figura 2.1 descreve todas as etapas de forma esquemática.

2.1.3 Stakeholders

Um **Engenheiro de Linguagem** tem como principal objetivo desenvolver linguagens que facilitem a criação de programas de Software, assim como criar um editor para a linguagem recém-criada, para que o utilizador do domínio o possa utilizar de forma eficaz. Na grande maioria dos casos, os Engenheiros de Linguagens estão envolvidos na especificação, implementação e avaliação da linguagem [Kle08].

Um **Utilizador da LDE** ou um **Utilizador do Domínio**, é a pessoa responsável por criar aplicações através da linguagem de Software [Kle08].

Um **Especialista no Domínio** é uma pessoa que se encontra envolvida no processo de desenvolvimento da linguagem. São especialistas num determinado domínio, não necessariamente em Informática, que precisam de usar ambientes informáticos para as suas atividades diárias [Cos+03]. São responsáveis por identificar o número de iterações necessárias, até atingir o produto final.

2.1.4 Paradigmas de Desenvolvimento

Sendo as LDEs apresentadas em modelos e meta-modelos, os paradigmas são definidos para determinar a importância desses elementos na construção de uma LDE. Então é possível definir um conjunto de paradigmas essenciais [Ame09]:

- **Model-Based Engineering (MDE)** - Engenharia Baseada em Modelos é o processo no qual os modelos desempenham um papel importante, mas não essencial, para o desenvolvimento do Software.
- **Model-Driven Engineering (MBE)** - Engenharia Orientada a Modelos vai além das atividades de desenvolvimento e concentra-se na criação e exploração de modelos de domínio.
- **Model-Driven Development (MDD)** - Desenvolvimento Orientado a Modelos é um paradigma de desenvolvimento, que utiliza modelos e transformações como o principal artefacto no processo de desenvolvimento de um Software. Na grande maioria dos casos, neste paradigma a implementação é gerada automaticamente a partir dos modelos.
- **Model-Driven Architecture (MDA)** - Arquitetura Orientada a Modelos é uma visão particular do Desenvolvimento Orientado a Modelos, desenvolvida pela *Object Management Group* (OMG). Possibilita a definição de aplicações e modelos de dados que permite a flexibilidade a longo prazo em relação à implementação, integração, manutenção, teste e simulação [Ngu+14]

2.1.5 Ferramentas de Desenvolvimento

Existem várias ferramentas que permitem o desenvolvimento de LDEs. Não se pretendendo apresentar uma lista exaustiva, é apresentada uma lista de algumas das ferramentas utilizadas:

- **Eclipse Modeling Framework (EMF)** - Ferramenta de desenvolvimento integrada com o Eclipse, permite a construção de meta-modelos Ecore e fornece também ferramentas de suporte para produzir um conjunto de classes Java. Através deste meta-modelo Ecore, o EMF permite também uma geração de código e uma validação do meta-modelo [Ste+08].

- **Sirius** - Ferramenta baseada no EMF, que simplifica a especificação do produto, reduz o tempo de design e aumenta rapidamente a produtividade do projeto. Fornece uma ferramenta que permite especificar diferentes pontos de vista no mesmo modelo, através da definição de conceitos e das suas relações. Depois de definido o modelo, permite também a criação de representações concretas desses modelos [Viy+14].
- **MetaEdit** - Ferramenta de modelação que permite criar e usar LDE. É baseada numa meta-linguagem GOPPRR (Grafo, Objeto, Propriedade, Porta, Relação, Regra), sendo que todos os conceitos são definidos independentemente uns dos outros e as suas relações são especificadas posteriormente quando todos os conceitos forem definidos [Spr10]. Esta ferramenta, tal como as referidas anteriormente, permite também uma geração de código e uma validação do meta-modelo.
- **Obeo** - Ambiente de modelação de LDE baseado em pontos de vista, sendo baseado no EMF. Tal como o MetaEdit são ferramentas comerciais, divididas em duas partes diferentes: um *workbench*, uma ferramenta para projetar LDE e um modelador, uma ferramenta para usar a LDE [Dus+13].

2.1.6 Plataformas Alvo

- **JADE** - Do inglês *Java Agent Development Framework*, facilita o desenvolvimento de aplicações multi-agente que se encontrem em conformidade com as especificações FIPA (*Foundation for Intelligent Physical Agents*). Usa um modelo de agente e tal como o nome indica é baseado na linguagem Java, o que permite uma boa eficiência no tempo de execução, reutilização de Software, mobilidade de agentes e a realização de diferentes arquiteturas de agentes [Bel+01].
- **JADEX** - Do inglês *Java Agent Development Framework eXtension*, é baseada na plataforma JADE e permite a criação de agentes seguindo a arquitetura *Belief-Desire-Intention* (BDI), mas adicionando novos conceitos como a definição de objetivos explícitos e permitir a agregação de novas funcionalidades ao nível da análise dos resultados [Bor+06].
- **JACK** - Modelo que estende a linguagem Java, e suporta um novo paradigma de programação, isto é, uma linguagem de programação orientada a agentes. Este modelo oferece um alto nível de performance, uma implementação leve e uma maneira fácil de ser estendida para dar suporte a diferentes modelos de agentes e requisitos específicos das aplicações [Nun07].

2.2 Usabilidade

Hoje em dia, a usabilidade é um dos aspetos mais importantes no processo de desenvolvimento de Software. Entre as várias definições para usabilidade, destaca-se a da norma

ISO 9241-11 [Bev+15], que se tem tornado como a principal referência em que é dito que a usabilidade é: "*The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*".

A usabilidade está diretamente relacionada com a facilidade que uma interface é de usar por parte de um utilizador. De acordo com Jakob Nielsen [Nie03], é possível definir 5 componentes principais de usabilidade:

1. **Aprendizagem** - Mede o quão fácil é a linguagem de aprender, quando os utilizadores usam o sistema pela primeira vez.
2. **Eficiência** - Após os utilizadores aprenderem a usar o sistema, mede a velocidade com que os utilizadores executam determinadas tarefas.
3. **Memorização** - Quando os utilizadores regressam ao sistema após um período sem o usar, mede o tempo que estes precisam para saber usar o sistema novamente;
4. **Erros** - Quantos erros os utilizadores cometem ao usar o sistema, se são erros graves ou não e a facilidade de recuperar desses erros.
5. **Satisfação** - Verifica se os utilizadores têm confiança e segurança no sistema.

Assim, para avaliar a usabilidade global de um sistema é necessário ter como base um conjunto de métricas de usabilidade, considerando os valores das métricas para avaliar qualitativamente e quantitativamente os atributos de um determinado sistema.

2.3 Notações Visuais

A memória humana é um sistema de armazenamento que preserva toda a informação pretendida. Esta pode ser dividida em quatro categorias [WW06]:

1. **Memória Sensorial** - Capacidade de guardar informações através do sistema sensorial humano, que é composto por audição, visão, olfato, paladar e tato. Estas informações são armazenadas por um curto espaço de tempo e caso sejam processadas passam então para a memória a curto prazo.
2. **Memória a curto prazo** - Capacidade de guardar informações já codificadas pelo sistema sensorial durante um curto período de tempo. Estas informações podem ser esquecidas ou passar para a memória de longo prazo.
3. **Memória a longo prazo** - Capacidade de guardar informações recebidas pela memória a curto prazo. Possui capacidade ilimitada de armazenamento.
4. **Memória de trabalho** - Capacidade de guardar informações enquanto nos for útil. É muitas vezes utilizada em atividades mentais, que exijam raciocínio.

O cérebro humano e os seus mecanismos de processamento, são essenciais na vida humana e para Engenharia de Software não é exceção. Para usufruir de todas as capacidades destes tipos de memórias, é necessário aplicar metodologias como as notações visuais. Essas notações funcionam como meio de comunicação visual eficaz e a escolha de uma notação errada, pode desencadear a transmissão de uma mensagem incorreta. De acordo com *Moody* [Moo+10], a eficácia cognitiva está relacionada com velocidade, facilidade e precisão com que a representação das notações visuais pode ser processada pela mente humana, sendo definido como o principal fator para comparar notações visuais.

2.3.1 Physics Of Notations

Moody no seu trabalho "*Physics Of Notations*" [Moo09], estabeleceu um conjunto de propriedades físicas e lógicas desejáveis para notações visuais, otimizadas para a comunicação humana e resolução de problemas, fornecendo assim uma base para o desenvolvimento e avaliação de notações de forma a aumentar a eficácia cognitiva. Estes princípios podem ser usados para avaliar, comparar e melhorar notações visuais existentes, bem como para construir novas.

Apesar de existirem um vasto conjunto de princípios para melhorar as propriedades de comunicação de uma determinada linguagem, nesta dissertação apenas foi avaliada a transparência semântica pelo que vai ser feita uma breve descrição da mesma.



Figura 2.2: Os 9 princípios de "Physics Of Notations". Adaptado de [Moo09].

2.3.1.1 Transparência Semântica

Transparência Semântica indica os símbolos devem sugerir o seu significado. Os símbolos podem ser:

1. **Semanticamente Imediato** - Um leitor consegue compreender o significado do símbolo através da sua aparência;
2. **Semanticamente Perverso** - Se um leitor consegue deduzir um significado diferente ou até mesmo oposto do símbolo;

3. **Semanticamente Translucido** - Se um leitor consegue compreender o significado do símbolo após uma explicação inicial.

2.4 Eye Tracker

Os dispositivos de *Eye Tracker* começaram a ser usados pela Engenharia de Software a partir de 1990 [Sha+15] e nos últimos anos têm-se tornado uma ferramenta cada vez mais útil para a realização de estudos empíricos, podendo ser usados para estudar a compreensão do modelo e a compreensão do código (Fig. 2.3).



Figura 2.3: Eye Tracker

Um *Eye Tracker* é sistema que permite recolher dados sobre o processo cognitivo de alguns utilizadores. Através desses dados, é possível realizar um mapeamento entre a posição dos olhos do utilizador e o local aproximado que está a observar, podendo ser recolhidos, entre outros, os seguintes indicadores:

1. **Fixação** - Momento em que o olho se encontra praticamente imóvel. Todos os *Eye Tracker* possuem estes dados que incluem as coordenadas cartesianas e a data/hora em que esse estímulo ocorreu.
2. **Sacada** - (do inglês *saccade*) Movimento ocular muito rápido, cerca de 40ms a 50ms, de uma posição de interesse para a outra.
3. **Caminho de Inspeção** - Caminho espacial e temporal de todas as sequências de fixações e sacadas que o utilizador realiza.

A figura 2.4, representa um exemplo de um caminho de inspeção de um utilizador, em que os círculos numerados indicam a fixação do utilizador, sendo que o tamanho é proporcional à duração do tempo e os números representam a ordem de visualização e as linhas representam o movimento ocular. Assim sendo, é possível verificar que a classe *NTupleController* foi observada antes da classe *PyNTupleController*, e que o círculo com maior área é o da classe *PyNTuple*, indicando que foi gasto uma grande quantidade de tempo a visualizar esta classe.

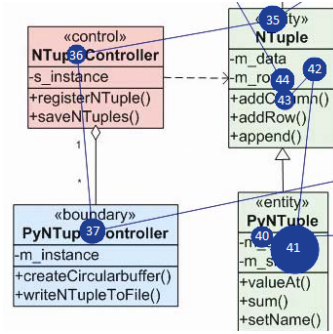


Figura 2.4: Exemplo de um Caminho de Inspeção. Adaptado de [Yus+07].

De acordo com Jacob [Jac90], o processamento de todas as informações ocorre durante o processo de fixação enquanto que pouco, ou mesmo nenhum, processamento de informação ocorre durante o período do movimento ocular (sacada).

2.4.1 Métricas

Depois de juntar todos os dados do teste do utilizador, é necessário aplicar um conjunto de métricas de forma a poder analisar quantitativamente o seu desempenho. De seguida, são apresentadas algumas métricas que podem ser usadas para avaliar a usabilidade de um sistema.

- Número de fixações em geral - Um maior número de fixações em geral indica uma pesquisa menos eficiente.
- Fixações por área de interesse - Um maior número de fixações numa área em particular, indica que essa área é mais visível ou mais importante para o utilizador, do que as outras áreas.
- Duração da Fixação - Uma maior duração de fixação numa determinada área, indica dificuldades em extrair informação ou que a área possui algo de muito interesse.
- Ciclo de Fixação - Usualmente é a soma de todas as durações de fixação de uma determinada área. É usado para efetuar uma comparação geral com todas as áreas e verificar quais as áreas que o utilizador dispensou mais tempo.
- Densidade Espacial de Fixação - As fixações concentradas numa pequena área, indicam que está a ocorrer uma pesquisa focada e eficiente, caso contrário e as fixações estejam espalhadas refletem numa busca ineficiente.
- Fixações Repetidas - Um número elevado de fixações depois da área já ter sido fixada, indica que essa área possui pouca visibilidade.
- Tempo para a primeira fixação no alvo - Quanto menor for o tempo para a primeira fixação, maiores são as propriedades de atenção que essa área possui.

- Percentagem de participantes que se fixam numa área de interesse - Se um baixo número de participantes se fixaram numa área que é de interesse, então essa área deve ser movida ou destacada.
- No alvo - Número de fixações no alvo a dividir pelo numero total de fixações. Quanto menor o valor, menor é a eficiência da pesquisa.

2.5 System Usability Scale

Os utilizadores podem realizar um conjunto de exercícios de avaliação, mas após este período de tempo, podem estar muito cansados especialmente se encontrarem problemas de usabilidade. De seguida, caso sejam expostos a um questionário longo, estes podem não estar totalmente concentrados e até mesmo não responder, o que implicaria uma agregação de dados insuficiente.

Devido ao elevado grau de exigência de avaliar a satisfação dos utilizadores com os sistemas, surgiu o **SUS** [Bro+96], um dos métodos mais conhecidos e simples para avaliar a usabilidade de um sistema. Este método deve ser aplicado depois de um teste de usabilidade mais qualitativo e consiste num conjunto de 10 perguntas, sendo que para cada uma delas o utilizador pode responder numa escala de 1 a 5, onde 1 significa discordo completamente e 5 concordo completamente. As respostas possíveis por parte do utilizador, estão representadas na figura 2.5

O questionário deve conter as seguintes perguntas [Bor+09]: i) Acho que gostava de usar este sistema com frequência. ii) Acho o sistema desnecessariamente complexo. iii) Acho o sistema fácil de usar. iv) Acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema. v) Acho que as várias funções do sistema estão muito bem integradas. vi) Acho que o sistema apresenta muitas inconsistências. vii) Acho que as pessoas aprenderão como usar o sistema rapidamente. viii) Acho o sistema muito complicado de usar. ix) Senti-me confiante ao usar o sistema. x) Precisava de aprender várias coisas novas antes de conseguir usar o sistema.

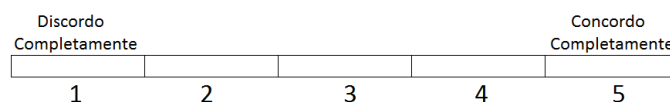


Figura 2.5: Respostas possíveis do utilizados ao questionário **SUS**.

2.5.1 Pontuação SUS

Depois de reunir todos os resultados, é preciso realizar um processo de forma a chegarmos à pontuação final. Para as perguntas 1,3,5,7 e 9, a pontuação é a posição da escala menos 1. Para as perguntas 2,4,6,8 e 10, é 5 menos a posição da escala. De seguida, somam-se todos

os valores e multiplicam-se por 2,5 de forma a obter um valor geral da usabilidade do sistema que pode ir de 0 a 100.

Embora o valor final esteja num intervalo de 0 a 100, este não deve ser considerado como uma percentagem. [Bro+96].

2.6 Web Semântica

"I have a dream for the Web in which computers become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A 'Semantic Web'" - Tim Berners-Lee.

A Web Semântica [BL+01], tem como objetivo principal tornar todas as informações que se encontram na Web mais acessíveis para os utilizadores, facilitando assim a vida dos mesmos. Nesse artigo é também dado um exemplo da Web Semântica, como um utilizador que acede a uma página da Internet de uma clínica e que para além de saber que a página possui palavras chave como "medicina, terapia, tratamento", sabe também a localização da clínica, informação sobre quais os médicos a darem consultas, a que dias de semana há consultas, etc. Neste sentido, caso o utilizador pretenda saber se um determinado médico dá consultas num determinado dia numa clínica perto de si, ao invés de fazer uma pesquisa por cada parâmetro, o que implicaria uma pesquisa em diversas páginas, o utilizador poderia fazer uma pesquisa do género: "Quero saber se o Dr. Rui Costa dá consultas dia 1 de Dezembro numa clínica perto de Viseu". A Web Semântica analisaria a pesquisa do utilizador, pesquisava na web combinações possíveis e de seguida organizava os resultados que se adequassem melhor ao utilizador.

A Web Semântica pode ser projetada para ser compreendida por máquinas, sob agentes computacionais, sendo estes capazes de atuar eficientemente sobre as informações. Isto só é possível através da integração de todos os dados na Internet, transformando-a numa base de dados aberta e compreensível pelas nossas máquinas.

2.7 Sistemas Multi-Agente

Um agente é uma entidade computacional, como um programa de Software ou um robô, que percebe e age no seu ambiente de forma autónoma. Este possui sensores de forma a ter uma perceção do meio ambiente e atuadores para agir nesse mesmo ambiente. Assim, o seu comportamento apenas depende da sua própria experiência [Wei99].

"An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future" - Stan Franklin Art Graesser.

Com o aumento do número de sistemas informáticos, os agentes são considerados um modelo de próxima geração sendo estes utilizados em diversas áreas desde a IPM até

processos de controlo industrial. Na Web Semântica um agente pode ser visto como uma entidade autónoma em que lhe são atribuídas determinadas tarefas.

SMA representam um conjunto de sistemas que interagem num ambiente, em que o objetivo principal é atingir um comportamento global inteligente a partir do comportamento individual dos agentes, de forma a desempenhar determinadas tarefas ou satisfazer um conjunto de objetivos.

O uso dessas linguagens, em vez de outras mais convencionais, é útil quando o problema é modelado como um **SMA** e entendido em termos de conceitos cognitivos e sociais, como crenças, metas, planos, papéis e normas [Bor+06]. Ao utilizar **SMA**, garantimos que o sistema a implementar é escalável pois é sempre possível adicionar novos agentes ao sistema. Sendo também reutilizável graças à sua modularidade, é possível reintegrar um agente num novo sistema. Para além disso, no caso da implementação de soluções que envolvam dados distribuídos, estas são facilmente implementadas pois no caso de um agente falhar é possível recuperar o sistema atribuindo a tarefa incompleta a outro agente.

A interação entre os agentes é um aspeto principal nos **SMA**, pois caso não houvesse uma interação entre dois ou mais agentes, era impossível estabelecer um comportamento global. Uma interação ocorre quando dois ou mais agentes são envolvidos numa relação dinâmica, através de um conjunto de ações que terão influência no comportamento futuro do agente. Esta interação pode ser feita diretamente entre os agentes, com um agente intermediário, ou até mesmo através do ambiente. Assim, os agentes cooperam entre si com o objetivo de realizar uma determinada tarefa [Sch01].

Agentes autónomos podem ser agrupados em vários tipos [Sch01]:

1. **Agentes Mentais** - Usam notações mentais que são aplicadas aos humanos. Possuem um modelo simbólico, mas explícito, do ambiente em que se encontra, usando dados de sensores para atualizar o seu modelo ambiente.
2. **Agentes Reativos** - Não possuem uma representação do seu ambiente, ao contrário dos agentes mentais. As suas decisões são tomadas com base em informações atuais visto não possuir um histórico do passado.
3. **Agentes Híbridos** - Junção dos dois agentes. Permite expressar uma estrutura do ambiente em que se encontra e permite também uma rápida reação a eventos sem a necessidade de manipulação de estruturas

2.7.1 Linguagens para **SMA**

O uso de linguagens de programação apropriadas para a implementação de um **SMA**, em vez de outras mais convencionais, é útil quando o problema é entendido em termos de conceitos cognitivos e sociais, como crenças, metas, planos, papéis e normas. Estas, podem ser categorizadas em 3 tipos: linguagens declarativas 2.7.1.1, linguagens imperativas 2.7.1.2 e linguagens híbridas 2.7.1.3.

2.7.1.1 Linguagens Declarativas

As linguagens de programação declarativa defendem um estilo de programação que expressa as propriedades dos problemas e das suas soluções, sendo caracterizadas pela sua forte natureza formal. Dependendo do formalismo subjacente para expressar tais propriedades, pode-se distinguir diferentes classes de linguagens declarativas, como linguagens de programação funcionais, lógicas ou de restrição [Pic+11].

De seguida, são apresentadas algumas linguagens declarativas para SMA:

- **CLAIM - Computational Language for Autonomous, Intelligent and Mobile Agents** - Linguagem de programação orientada a agentes declarativos de alto nível. Combina as principais vantagens das linguagens de programação orientadas a agentes, para representar aspetos cognitivos e raciocínio, com as linguagens concorrentes baseadas na álgebra de processos, para representar a simultaneidade e a mobilidade do agente. Esta linguagem é suportada por uma plataforma distribuída denominada por *SyMPA*, implementada em Java, que oferece todos os mecanismos necessários para gerir agentes, comunicação, mobilidade, segurança e tolerância a falhas.
- **MINERVA** - Sistema de agentes projetado para fornecer uma estrutura de agente comum baseada na programação lógica, para permitir a combinação de vários mecanismos de representação e raciocínio. Usa *MDLP - Multi-Dimensional Dynamic Logic Programming* e *KABUL - Knowledge And Behavior Update Language* para especificar agentes e os seus comportamentos. *MDLP* é um mecanismo básico de representação de conhecimento de um agente no *MINERVA* e *KABUL* é uma linguagem de programação lógica que permite a especificação de atualizações de conhecimentos. Assim, um agente *MINERVA* consiste em vários sub agentes especializados, possivelmente simultâneos, que executam várias tarefas, cujo comportamento é especificado em *KABUL*, enquanto lê e manipula uma base de conhecimento comum especificada no *MDLP*.

2.7.1.2 Linguagens Imperativas

Abordagens imperativas na programação orientada a agentes são menos comuns, principalmente devido ao fato de que a maioria das abstrações relacionadas ao design orientado a agentes é, normalmente, de natureza declarativa. No entanto, existem muitos programadores que ainda usam linguagens imperativas convencionais, isto é, não orientadas a agentes, para o desenvolvimento de SMA [Bor+06].

De seguida, é apresentada uma linguagem imperativa para SMA:

- **JACK Agent Language (JAL)** - Extensão do Java implementando alguns recursos de linguagens lógicas, como variáveis lógicas. São adicionadas várias construções sintáticas ao Java, permitindo que os programadores criem planos e bases de crenças, tudo de maneira gráfica, já que a JAL possui um IDE sofisticado que fornece uma

ferramenta para tal finalidade. Na JAL, os planos podem ser compostos por métodos de raciocínio e agrupados em capacidades. Outro mecanismo presente na JAL é a capacidade de usar equipas de agentes, ou organizações de agentes, uma noção que é cada vez mais importante no design orientado a agentes.

2.7.1.3 Linguagens Híbridas

Abordagens que combinam recursos declarativos e imperativos de modo a fornecer aos programadores abordagens flexíveis no desenvolvimento de SMA [SJ11].

De seguida, são apresentadas algumas linguagens híbridas para SMA:

- **Jason** - Permite a modelação de agentes para SMA num modelo lógico BDI. É uma versão estendida da *AgentSpeak*, uma linguagem de programação orientada a agentes.
- **3APL** - Permite a programação de agentes numa arquitetura cognitiva. Uma das principais características desta linguagem é o facto de conter construções de programação para implementar atitudes mentais, como crenças, objetivos, planos de agentes, bem como o processo de deliberação que os manipula. Suporta a integração de linguagem declarativa (Prolog) e linguagem imperativa (Java) [SJ11].

AMBIENTES DE DESENVOLVIMENTO PARA SISTEMAS MULTI-AGENTE

O interesse por [SMA](#) tem vindo a crescer nos últimos anos, pelo que é necessário o uso de ferramentas que dêem suporte ao desenvolvimento desses sistemas, pois criar e manipular dados sem o apoio de uma plataforma adequada, são tarefas muito complicadas.

De seguida são apresentadas algumas plataformas para o desenvolvimento de [SMA](#), que são similares ao [SEA_ML](#).

- **SEA_ML** - [SEA_ML](#) é uma [LDE](#), que permite modelar e gerar artefactos para uma arquitetura [SMA](#) [[Cha+14](#)]. Esta linguagem, visto ser a linguagem a ser estudada nesta dissertação, será apresentada com maior detalhe no capítulo 8.0.2.
- **MAS-ML** - *Multi-Agent Systems Modeling Language* (MAS-ML), é uma linguagem de modelação cujo principal objetivo é oferecer extensões ao meta-modelo UML, com base no [Taming Agents and Objects \(TAO\)](#)¹, para incorporar novos recursos para modelação de agentes. Esta linguagem permite especificar agentes, organizações e ambientes [[DS+08](#)].
- **ANote** - Linguagem de Modelação para [SMA](#), que permite especificar objetivos, agentes, contextos, ontologias e comportamentos. Esta linguagem apresenta também sete pontos de vista: visão do objetivo, visão do agente, visão ontológica, visão do cenário, visão do planeamento, visão de interação e visão de organização [[CL05](#)].

¹Framework que permite uma abstração de agentes e objetos

3.1 SEA_ML

SEA_ML é uma LDE, que inclui um conjunto de ferramentas gráficas, para modelar e gerar SMA. Esta linguagem, como é baseada numa arquitetura MDD, para além de permitir aos utilizadores modelar sistemas de agentes num nível independente da plataforma, permite também obter automaticamente ficheiros necessários para a execução do sistema [Cha+14].

3.1.1 Sintaxe Abstrata

Com o objetivo de proporcionar uma compreensão clara e um uso eficiente da linguagem, o meta-modelo SEA_ML é dividido em oito pontos de vista [Cha+14], onde cada uma descreve diferentes aspetos da web semântica para SMA.

1. **SMA** - Visão geral do meta-modelo, composta por **Semantic Web Organization (SWO)** que interage num ambiente. Um SWO é constituído por **Semantic Web Agents (SWA)** e cada organização pode ser composta por várias sub organizações de forma recursiva.
2. **Agent Internal** - Visão que se concentra na estrutura interna de cada agente que se encontra dentro de uma organização SMA. Estes agentes usam recursos do SWO no qual trabalham e podem desempenhar papéis e usar ontologias de forma a manter os seus conhecimentos com o objetivo de resolver os seus problemas.
3. **Plan viewpoint** - Quando um agente aplica um plano, ele precisa de executar as suas ações para completar as tarefas. Assim, cada ação precisa de estar conectada a uma entidade mensagem, que possui um remetente e um recetor, que as envia e recebe. Estas ações são atómicas, isto é, são executadas completamente em caso de sucesso ou abortadas em caso de erro.
4. **Role viewpoint** - SWA e SWO podem cumprir papeis e usar ontologias para manter o seu conhecimento interno e inferir sobre o meio ambiente com base nos fatos conhecidos. Esta visão é uma entidade do modelo geral que deve ser especializada no meta-modelo de acordo com tarefas arquitetónicas e de domínio.
5. **Interaction viewpoint** - Visão que se concentra nas comunicações e interações de um agente num SMA e define as entidades e relações como Interação, Mensagem e Sequência de Mensagens. Os agentes interagem uns com os outros e cada interação consiste num envio de mensagens, sendo que cada uma deve ter um tipo de mensagem (*msgtype*) como informação, solicitação ou confirmação.
6. **Environment viewpoint** - Concentra-se nas relações entre os vários agentes e a que informação é que eles podem aceder. O ambiente no qual estes agentes residem, contém todos os recursos e serviços que são utilizados pelos agentes. Cada serviço pode

ser um serviço na Internet ou outro serviço com protocolo de invocação predefinido na implementação da vida real. Sendo os factos baseados no meio ambiente, estes podem mudar ao longo do tempo caso o ambiente tenha novos conhecimentos de recursos diferentes. Cada agente pode continuar a acreditar na sua própria crença embora os factos tenham sido mudados, caso não atualize o seu conhecimento, o que pode levar a um conhecimento inconsistente em relação aos factos do mundo real.

7. **Agent-SWS interaction viewpoint** - Esta visão modela a interação entre os agentes e os [Semantic Web Services \(SWS\)](#) e preocupa-se com os serviços e acordos que devem ser realizados, de modo a suportar comunicações bem sucedidas. [SMA](#) e [SWS](#) são sistemas independentes. No entanto, podem interagir uns com os outros de modo a realizar determinadas tarefas com sucesso. É possível modelar o serviço de um agente como um [SWS](#), sendo que a ferramenta irá gerar todos os ficheiros necessários.
8. **Ontology viewpoint** - Esta visão engloba todos os conceitos de ontologia. Ontologia é uma forma de representação do conhecimento sobre o mundo, neste caso para membros do [SMA](#). Uma coleção de ontologias cria uma base de conhecimento do [SMA](#) sobre o contexto do domínio, sendo que no [SEA_ML](#) essas ontologias são representadas como instâncias de *OrganizationOntology*.

3.1.2 Sintaxe Concreta

O [SEA_ML](#) fornece ao utilizador um conjunto de construções semânticas, apresentadas através de uma notação visual em oito pontos de vistas diferentes.

A Tabela 3.1 apresenta a sintaxe concreta dos objetos usados no [SEA_ML](#).

3.1.3 Exemplo: Troca de Carros

Neste caso de estudo denominado por "Troca de Carros", um permutador pretende efetuar uma troca de carros. Para tal, o permutador tem acesso a um conjunto de informações sobre carros sendo que irá precisar delas para decidir se executa o plano *ExchangerFinderPlan*. Supõe-se que existe um serviço web para efetuar essa troca de carros.

De seguida, são apresentados três pontos de vistas: *Agent* (Fig. 3.1), *MAS* (Fig. 3.2) e *SWS-Interaction* (Fig. 3.3).

Conceito	Notação	Conceito	Notação
Semantic Web Agent		Service Ontology	
Capability		Action	
Role		Receive	
Belief		Send	
Goal		Task	
Plan		Domain Role	
Agent State		Registration Role	
Agent Type		Interface	
Behavior		Process	
Environment		Grounding	
Fact		Input	
Resource		Output	
Service		Pre-Condition	
Interaction		Effect	
Message		Semantic Service Register Plan	
Message Sequence		Semantic Service Finder Plan	
Semantic Web Organization		Semantic Service Agreement Plan	
Ontology Mediator Role		Semantic Service Executor Plan	
Organization Ontology		Web Service	
Role Ontology		ODMOWLClass	
Semantic Web Service		Architecture Role	
Semantic Service Matchmaker Agents			

Tabela 3.1: Sintaxe Concreta *SEA_ML*.

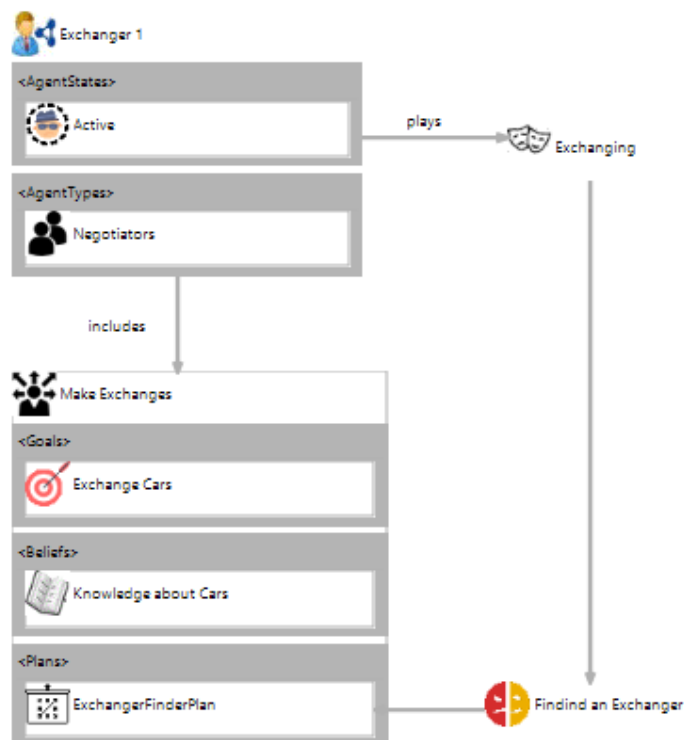


Figura 3.1: Ponto de Vista *Agent* para o exemplo Troca de Carros.

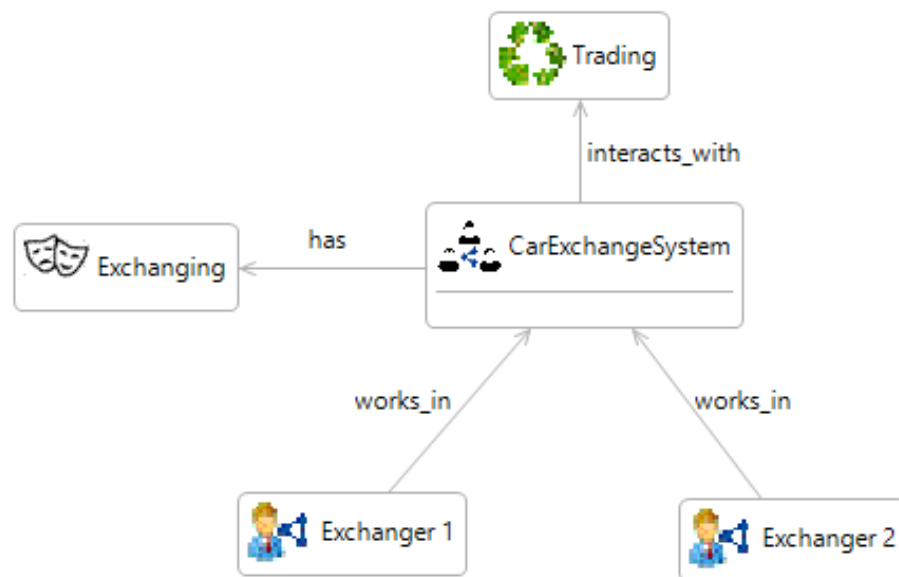


Figura 3.2: Ponto de Vista MAS para o exemplo Troca de Carros.

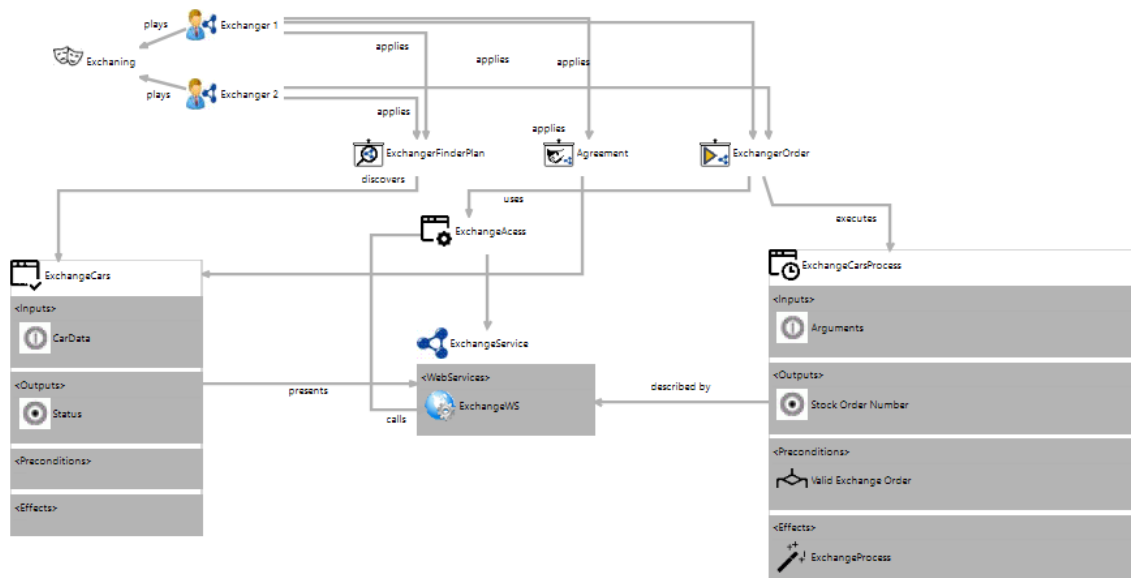


Figura 3.3: Ponto de Vista SWS-Interaction para o exemplo Troca de Carros.

3.2 DSML4MAS

DSML4MAS, é uma linguagem de modelação para domínio específico de **SMA**, que se baseia nos princípios de desenvolvimento orientado a modelos e permite modelar sistemas de agentes de forma independente e com recurso a uma plataforma gráfica [Hah08].

Tal como o **SEA_ML**, as transformações do modelo para uma linguagem de programação do agente JACK e JADE, também são possíveis com o **DSML4MAS**.

3.2.1 Sintaxe Abstrata

A sintaxe abstrata do **DSML4MAS** é definida através do meta-modelo PIM4AGENTS, uma linguagem que utiliza a abordagem **MDD** que pode ser usada para modelar sistemas de agentes de forma abstrata, baseando-se no Ecore, um meta-modelo do **EMF**.

O **DSML4MAS** encontra-se estruturado em sete pontos de vista importantes [Art+09]:

1. **Agent** - Define como modelar entidades autónomas através das capacidades que eles têm para resolver determinadas tarefas e os papéis que desempenham no **SMA**. Permite também definir que recursos a quem um agente tem acesso e qual o tipo de comportamento que ele pode ter para as resolver.
2. **Role** - Define quais as funções desempenhadas por cada agente dentro de uma organização. Os recursos utilizados podem incluir permissões, responsabilidades e protocolos.
3. **Organization** - Define como os agentes se encontram estruturados em organizações mais complexas.
4. **Collaboration** - Especifica a interação entre o subconjunto das funções de uma organização.
5. **Interaction** - Define como são feitas as trocas de mensagens entre os agentes. Pode ser descrita a partir da perspetiva interna de cada agente ou em termos de protocolos da interação entre agentes.
6. **Behavior** - Define o vocabulário disponível para especificar o comportamento interno dos agentes. Esse vocabulário pode ser definido em termos de ações simples ou mais complexas, usadas para alcançar os seus objetivos.
7. **Deployment** - Descreve as instâncias do agente em tempo de execução que se encontram envolvidas no sistema e como são atribuídas às funções da organização.

3.2.2 Sintaxe Concreta

O **DSML4MAS** fornece ao utilizador um conjunto de construções semânticas, apresentadas através de uma notação visual em oito pontos de vistas diferentes.

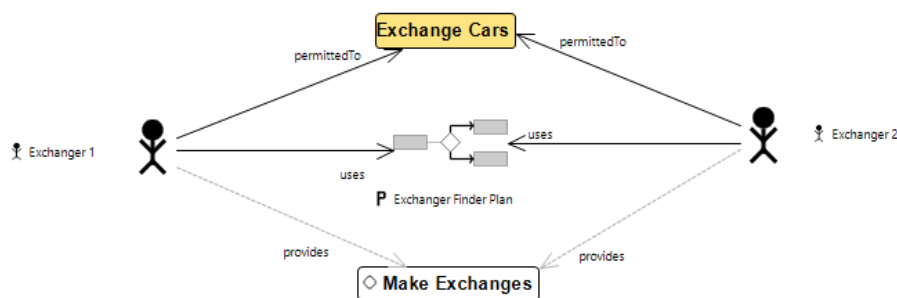
A Tabela 3.2 apresenta a sintaxe concreta dos objetos usados no **DSML4MAS**.

Conceito	Notação	Conceito	Notação
Agent		OECondition	
Plan		Precondition	
Capability		Effect	
DomainRole		Send	
Knowledge		Receive	
Collaboration		InternalTask	
DomainRoleBinding		Wait	
ActorBinding		Sequence	
Protocol		Decision	
Actor		Loop	
AgentInstance		Parallel	
OrganizationInstance		ParallelLoop	
link / MemberShip / hasParameter / hasOutput / controlFlow / informationFlow		Reply	
Attribute		AssignRole	
inheritsFrom		InitiateProtocol	
Input		InvokeWS	
Output		ReceiveWS	
Shared		MessageFlow	
DomainRole		MessageScope	
Interaction		ACLMessage	
OntologyMediatorRole		Break	
Timeout		Advertises	
SemanticWebService		Realizes	

Tabela 3.2: Sintaxe Concreta DSML4MAS.

3.2.3 Exemplo: Troca de Carros

De seguida, são apresentados três pontos de vistas para o mesmo exemplo apresentado na linguagem *SEA_ML* (3.1.3), mas agora para a linguagem *DSML4MAS*: *Agent* (Fig. 3.4), *MAS* (Fig. 3.5) e *Organization* (Fig. 3.6).

Figura 3.4: Ponto de Vista *Agent* para o exemplo Troca de Carros.

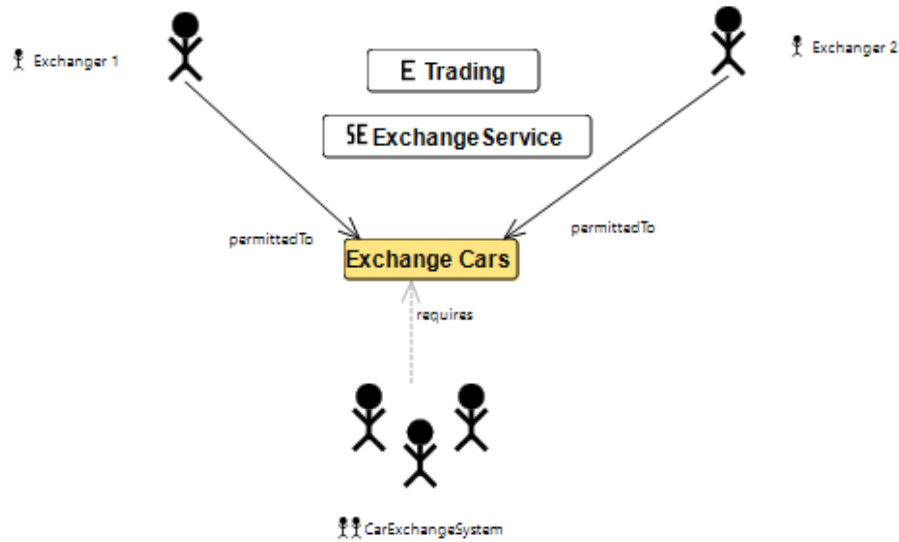


Figura 3.5: Ponto de Vista MAS para o exemplo Troca de Carros.

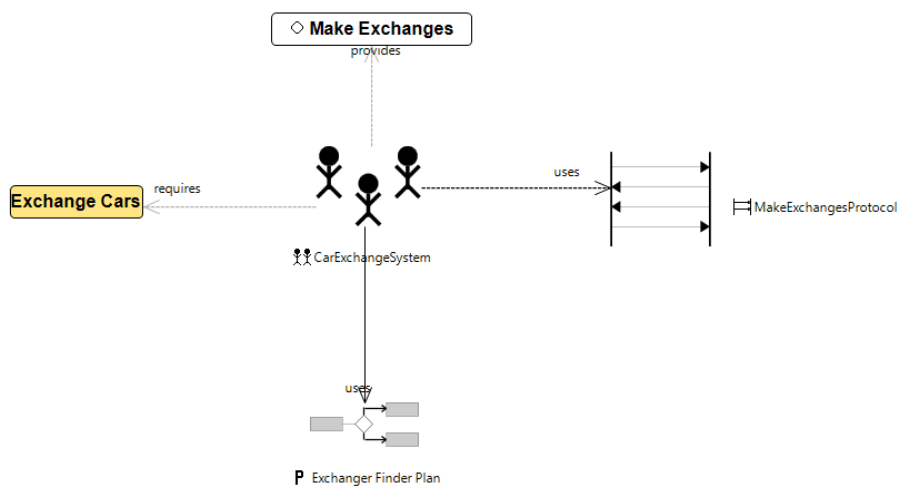


Figura 3.6: Ponto de Vista *Organization* para o exemplo Troca de Carros.

TRABALHOS RELACIONADOS

4.1 Abordagem para Desenhar e Avaliar a Eficácia Cognitiva de Sintaxes Concretas

A Engenharia de Requisitos estabelece um processo de identificação de todos os requisitos e documentos envolvidos, de forma a serem gerados de maneira apropriada para o seu design e implementação. O seu sucesso está relacionado com uma comunicação efetiva entre os Especialistas do Domínio e os seus utilizadores finais.

Um estudo realizado por *Caire et al.* [Cai+13], propõe uma nova abordagem para desenhar e avaliar a eficácia cognitiva de sintaxes concretas de Linguagens de Modelação de Requisitos envolvendo *stakeholders*. Neste artigo, é proposto uma abordagem inovadora para conceber notações visuais para engenharia de requisitos, sendo usado um conjunto de utilizadores não experientes no processo. É usado o *i**, uma das notações para Engenharia de Requisitos mais influentes.

Para avaliar a eficácia cognitiva do *i**, foram realizados 5 estudos empíricos, mas será apenas observada a avaliação da transparência semântica visto estar diretamente relacionado com a avaliação a realizar nesta dissertação.

Nesta experiência, foi avaliada a capacidade que os utilizadores não experientes têm de inferir o significado dos símbolos. Foi utilizado um método de interpretação cega, para medir a compreensibilidade dos símbolos gráficos. A essência deste tipo de teste é que são mostrados aos participantes os símbolos e é pedido para adivinhar ou inferir o significado dos mesmos. A compreensão do símbolo normalmente é medida pela percentagem de respostas corretas (taxa de sucesso). As principais perguntas de investigação para a avaliação da transparência semântica são: RQ1) Os utilizadores novatos conseguem desenhar símbolos mais semanticamente transparentes do que especialistas? RQ2) Será que a melhoria da transparência semântica melhora a compreensão dos utilizadores novatos?

4.1.1 Processo de Desenvolvimento do Estudo

Para avaliar a RQ1, foram utilizados 65 participantes que não tinham quaisquer conhecimentos sobre o i^* . Havia quatro grupos experimentais, cada um correspondendo a diferentes níveis do i^* :

- **Standard i^*** - Produzido por especialistas usando a intuição.
- **PoN i^*** - Produzido por especialistas usando princípios explícitos.
- **Stereotype i^*** - Símbolos mais comuns produzidos por novatos.
- **Prototype i^*** - Melhores símbolos produzidos por novatos, que foram testados por outros utilizadores também novatos.

Foram preparadas 4 experiências, uma para cada conjunto de símbolos. Para cada experiência, a primeira página foi usada para serem feitas algumas perguntas de triagem e as restantes para avaliar a transparência semântica dos símbolos. Cada símbolo era apresentado na parte superior da página e o conjunto completo das notações do i^* e as suas definições foram exibidas numa tabela abaixo. Os participantes foram distribuídos aleatoriamente entre grupos, e foi fornecido uma cópia do material experimental. De seguida, tinham de fazer uma correspondência entre o símbolo apresentado na parte superior da página e um símbolo apresentado na tabela. Para garantir algum efeito de sequência, tanto a ordem em que os símbolos foram apresentados como a ordem dos símbolos na tabela, foram alterados em cada página.

Para avaliar a RQ2, foram utilizados 83 participantes que também não possuíam quaisquer conhecimentos sobre o i^* . Os grupos utilizados foram os mesmos da experiência anterior, com mais um grupo adicional: PoN com desenho racional (PoN DR).

Foram preparados 5 conjuntos e materiais, um para cada grupo:

- **Materiais de Treino** - Definiram todos os símbolos e significados associados para um conjunto de símbolos.
- **Materiais de Teste** - Foram utilizados para avaliar a habilidade dos participantes em reconhecer com precisão os símbolos.

Foi apresentado os materiais de treino aos participantes, para que entendessem todos os símbolos e os seus significados. Nesta fase os participantes não podiam tomar qualquer tipo de notas. De seguida, passaram para a fase de teste onde os símbolos foram apresentados um por página e os participantes tinham de identificar o conceito correspondente.

4.1.2 Análise dos Resultados

Para analisar os resultados obtidos, foram utilizadas algumas métricas como a taxa de sucesso, que indica a percentagem de respostas certas, e o coeficiente de transparência semântica que indica o grau de aproximação entre um símbolo e a sua construção semântica.

$$\text{Taxa de Sucesso} = \frac{\text{Número de respostas certas}}{\text{Número total de respostas}}$$

$$\text{Coeficiente de Transparência Semântica} = \frac{\text{Frequência máxima} - \text{Frequência esperada}}{\text{Número total de respostas} - \text{Frequência esperada}}$$

$$\text{Frequência máxima} = \text{Número de respostas certas}$$

$$\text{Frequência esperada} = \frac{\text{Número total de Respostas}}{\text{Número total de Símbolos}}$$

Para a RQ1 foi possível concluir que os símbolos gerados por novatos são mais semanticamente transparentes do que os gerados por especialistas. A transparência semântica média dos símbolos gerados por novatos foi 5 vezes maior do que a dos símbolos gerados por especialistas. Para a RQ2, conclui-se que a transparência semântica aumenta significativamente a precisão do reconhecimento e reduz erros de interpretação.

4.2 Abordagem Sistemática para avaliar ambientes de linguagem de domínio específico para sistemas multi-agente

Um estudo realizado na *Ege University International Computer Institute* - Turquia [Cha+16], teve como principal objetivo a criação de uma estrutura de avaliação e uma abordagem sistemática para a avaliação de LDEs existentes ou recém-criadas para SMA. Para tal, foi usado o SEA_ML para demonstrar a abordagem de avaliação.

4.2.1 Processo de Desenvolvimento do Estudo

Para avaliar o desenvolvimento da linguagem SEA_ML, foram escolhidos dois grupos de utilizadores onde desenvolveram SMA para quatro estudos diferentes. O primeiro grupo usou o SEA_ML, enquanto os membros do segundo grupo, chamados programadores manuais, não utilizaram qualquer LDE e eram livres de escolher qual a LPG usar.

Para avaliar a LDE do SMA foi criado um processo trifásico que inclui as seguintes fases:

1. **Preparação** - Nesta fase é analisado o domínio, é criado um exemplo de forma a abranger as dimensões especificadas, são selecionados alguns utilizadores, integrados em duas equipas diferentes e no último passo é feita uma apresentação do

[SEA_ML](#) à equipa que a irá usar, de forma a puderem usar a linguagem para a avaliação.

2. **Execução** - Inicialmente é feita uma apresentação sobre o exemplo a criar e de seguida as equipas desenvolvem esse mesmo exemplo. No final, é apresentado um questionário aos utilizadores e são guardados todos os dados.
3. **Análise** - São extraídas as informações dos resultados qualitativos e quantitativos e é feita uma avaliação da ferramenta de forma adequada, com tabelas e diagramas. Finalmente é feito um relatório de avaliação detalhado.

4.2.2 Análise dos Resultados

Através dos resultados obtidos foi possível concluir que o uso de uma [LDE](#) para [SMA](#), pode reduzir a complexidade do desenvolvimento. Para chegarem a essa conclusão, foi feita uma análise dos resultados para os dois grupos, sendo o grupo A o conjunto de utilizadores que utilizou o [SEA_ML](#) e o grupo B os utilizadores que não utilizaram qualquer tipo de [LDE](#). Então, foi possível chegar as seguintes conclusões:

1. A análise do problema faz parte do desenvolvimento de qualquer sistema de software, pelo que ambos os grupos gastaram aproximadamente o mesmo tempo.
2. O tempo de modelação é próximo em ambos os grupos, mas o grupo A é um pouco mais rápido, pois poderia usar [SEA_ML](#) e trabalhar com conceitos específicos do domínio, enquanto o grupo B cumpriu este passo sem usar o [SEA_ML](#).
3. Na fase da implementação é que se destaca uma grande variação de tempo, sendo que o tempo gasto na implementação no grupo B é muito maior do que no grupo A. O motivo pelo qual a diferença é tão grande é que o grupo A utiliza o [SEA_ML](#) enquanto o grupo B utiliza uma [LPG](#).
4. Na fase de deteção de erros, o tempo gasto pelo grupo A é inferior ao do grupo B. Isto deve-se ao facto de o grupo B ter de encontrar erros em todo o código, enquanto o grupo A apenas se concentra em encontrar erros no código complementar adicionado que é muito inferior ao código gerado.
5. Foi também pedido que ambos os grupos efetuassem uma extensão para cada caso de estudo. O grupo A usa a [LDE](#) para essa extensão, pelo que adicionam ou modificam os modelos de instância e geram de novo o código, enquanto no grupo B essa modificação é feita manualmente. Assim sendo, o tempo gasto pelo grupo A é inferior ao do grupo B.

Assim, depois de efetuada esta análise, foi possível verificar que o tempo total de desenvolvimento do grupo A é metade do tempo do grupo B.

4.3 Avaliar a Framework SEA_ML

Um estudo realizado na Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa em conjunto com o Instituto Internacional de Informática da *Ege Univeristy* [Mir17], teve como principal objetivo avaliar a usabilidade de quatro LDE para SMA utilizando os princípios do *Physics Of Notations* (ver secção 2.3.1) de forma a poder comparar o impacto entre a notação visual antiga e a notação visual proposta.

4.3.1 Processo de Desenvolvimento do Estudo

Inicialmente, foram aplicados cada princípio do *Physics Of Notations* no SEA_ML, de forma a verificar se é possível melhorar a notação visual da linguagem. Depois disto, foi proposta uma melhoria para a notação visual, denominada por SEA_ML++, e foram desenvolvidas um conjunto de duas experiências, com recurso à framework USE-ME, com participantes que possuíam pouco ou até mesmo nenhum conhecimento sobre LDE e SMA. A framework USE-ME desenvolvida na NOVA-LINCS, contempla o processo de desenvolvimento incremental iterativo das LDEs sobre a questão da avaliação da usabilidade. O processo realizado para as duas experiências foi o seguinte:

1. Foram apresentados aos utilizadores teste, uma lista com as notações da SEA_ML e da SEA_ML++, juntamente com um conjunto de descrições textuais. Esta experiência teve uma duração de 30 minutos. O objetivo principal era verificar se os utilizadores selecionavam mais notações visuais da nova notação proposta do que da original.
2. Foram apresentados aos utilizadores teste quatro exercícios diferentes, que representam quatro pontos de vista da LDE, de forma a poderem testar a notação da SEA_ML e da SEA_ML++. Depois de cada exercício, era solicitado que o utilizadores respondessem a um questionário SUS (ver secção 2.5), de forma a obter uma apreciação sobre o exercício que tinham resolvido. Após terminarem os quatro exercícios, era apresentado um questionário sobre o perfil do utilizador. Visto que os utilizadores não tinham experiência, o principal objetivo era testar o grau de aprendizagem do SEA_ML.

4.3.2 Análise dos Resultados

Para a análise dos resultados, foram verificados vários fatores nas experiências como o tempo gasto numa determinada tarefa, taxa de erro, análise do erro, taxa de sucesso e satisfação do utilizador.

Depois da aplicação dos princípios do *Physics Of Notations* e das duas experiências efetuadas, foi verificado que é possível melhorar a sintaxe concreta do SEA_ML. Através dos resultados obtidos concluiu-se que apesar de na primeira experiência os utilizadores selecionarem um maior número de notações do SEA_ML++ em comparação com o SEA_ML,

na segunda experiência quando os utilizadores eram postos em prática, demoravam mais tempo a realizar o exercício com o [SEA_ML++](#) do que com o [SEA_ML](#).

4.4 Análise Comparativa

Através dos trabalhos investigados, é possível verificar que existem algumas limitações nas avaliações diretamente relacionadas com a linguagem [SEA_ML](#). Na avaliação da framework [SEA_ML](#) [Mir17], foi efetuada uma avaliação que se concentrou apenas nas notações da linguagem baseando-se nos princípios do *Physics Of Notations*. Apesar da sintaxe concreta da linguagem ser um fator que pode levar a linguagem a melhor qualidade do ponto de vista da usabilidade, existem outros fatores que devem ser tidos em conta quando se efetua uma avaliação de uma linguagem. Outra limitação nesta avaliação, foi o facto de os utilizadores teste serem inexperientes com [LDE](#) e [SMA](#), o que implica que não há garantias que os resultados fossem semelhantes com utilizadores experientes, um problema que ameaça a validade externa dos resultados. Assim, é necessária uma análise mais profunda, diretamente no contexto de uso da linguagem e que tenha em conta outros fatores de usabilidade, como por exemplo, a expressividade da linguagem.

Em relação à Abordagem Sistemática para avaliar [LDE](#) para [SMA](#), apesar de ser referido que é possível adotar esta abordagem para efetuar uma comparação entre duas [LDE](#), foi apenas efetuada uma comparação entre uma [LDE](#) e uma [LPG](#). Além disso, neste estudo foram comparadas algumas características, tendo apenas em conta o tempo gasto para realizar uma determinada tarefa, mas a usabilidade global de um sistema é determinada com base num conjunto de medidas de usabilidade - métricas e não apenas no tempo despendido.

Assim sendo, a avaliação a desenvolver sobre a linguagem [SEA_ML](#) apresenta diversas características que a destacam das referidas anteriormente. A avaliação da linguagem tem em conta outros fatores de usabilidade para além das notações visuais. Da mesma forma, possibilita a aplicação de um conjunto de métricas de usabilidade em contexto de uso, ao invés de se basear apenas no tempo para concluir uma determinada tarefa.

Para a avaliação da transparência semântica, será utilizado um processo semelhante ao desenvolvido por *Caire et al.* [Cai+13], desta vez comparando a notação original do [SEA_ML](#) com a notação proposta por [Mir17].

PREPARAÇÃO DA AVALIAÇÃO EXPERIMENTAL

5.1 Análise dos Objetivos

Como já referido anteriormente, nesta dissertação será feita uma avaliação da transparência semântica da linguagem, aspeto que ficou por avaliar na primeira iteração efetuada à linguagem, e de seguida será feita uma comparação com outra [LDE](#), mais especificamente o [DSML4MAS](#), de forma a identificar problemas que possam estar presentes relativamente à forma como o [SEA_ML++](#) se encontra estruturado. Foi escolhido o [DSML4MAS](#) como elo de comparação, pois ambas as [LDE](#) permitem modelar [SMA](#) de acordo com vários pontos de vista internos do agente e, para além disso, ambas as linguagens estão equipadas com IDEs baseadas em Eclipse, sendo que a modelação e a sua geração automática dos vários componentes é semelhante.

Nesta dissertação, foi utilizada a metodologia do [Goal Question Metric \(GQM\)](#) [CR94]. O [GQM](#) é uma estrutura hierárquica sendo um sucesso pois é adaptável a muitas organizações e ambientes diferentes, permitindo medir o desempenho em tarefas do processo de desenvolvimento de software. Este modelo de medição é dividido em três níveis: conceptual, operacional e quantitativo, sendo o nível conceptual os objetivos a atingir, o nível operacional o conjunto de perguntas que é usado como forma de atingir um objetivo e o nível quantitativo que é composto por métricas que permitem responder a cada questão de forma quantitativa.

Para ser feita a comparação, serão realizados um conjunto de estudos nos quais os participantes terão de efetuar algumas tarefas em ambas as linguagens de modelação e para a avaliação de cada componente de usabilidade, será utilizado um conjunto de métricas de forma a poder analisar quantitativamente e qualitativamente o desempenho do participante. De acordo com Dumas e Redish [DR99], um teste de usabilidade precisa de envolver necessariamente um conjunto de medidas quantitativas, como a satisfação do

participante ao longo do exercício, número de respostas erradas ou o tempo que demoram para completar uma tarefa.

De acordo com a [ISO 9241-11](#), a usabilidade é definida como a capacidade que um determinado produto tem de possibilitar aos utilizadores atingir os seus objetivos com eficácia, eficiência e satisfação. É também possível medir a usabilidade tendo em conta outras perspetivas, como a facilidade de aprendizagem, memorização e a baixa taxa de erros. O objetivo principal é avaliar a usabilidade do [SEA_ML++](#), sendo que este objetivo se divide em seis sub-objetivos:

- **Avaliar a eficácia** - Verificar se o exercício resolvido cumpre o objetivo proposto.
- **Avaliar a eficiência** - Verificar se o participante consegue realizar o exercício dentro de uma determinada unidade de tempo.
- **Avaliar a satisfação** - Verificar o grau de satisfação do participante ao realizar o exercício.
- **Avaliar a facilidade de aprendizagem** - Verificar se o sistema é fácil de aprender ou não.
- **Avaliar a memorização** - Avaliar a transparência semântica.
- **Avaliar a taxa de erros** - Verificar se o sistema é propício a erros ou não.

Para cada sub-objetivo é necessário definir um conjunto de questões de forma a tornar o objetivo principal mais específico e sustentar cada pergunta a uma ou mais métricas, permitindo assim quantificar o grau de satisfação do objetivo. Assim é possível definir a nossa estrutura como:

- **Avaliar a usabilidade do [SEA_ML++](#).**
 - **Objetivo** - Avaliar a Eficácia do [SEA_ML++](#)
 - * **Pergunta:**
 1. **RQ1:** Será que o exercício resolvido usando o [SEA_ML++](#) se encontra mais próximo do objetivo pretendido no caso de teste, em comparação com o [DSML4MAS](#)?
 - * **Métrica** - Número de tarefas completas/não completas e número de tarefas realizadas corretamente/incorrectamente.
 - **Objetivo** - Avaliar a Eficiência do [SEA_ML++](#)
 - * **Pergunta:**
 1. **RQ2:** Será que desenvolver um sistema no [SEA_ML++](#) requer menos esforço em comparação com o [DSML4MAS](#)?
 - * **Métrica** - Tempo gasto em ambas as linguagens e dados obtidos pelo *Eye Tracker*.

- **Objetivo** - Avaliar a Satisfação do [SEA_ML++](#)
 - * **Pergunta:**
 1. **RQ4:** Será que com o uso do [SEA_ML++](#) o utilizador apresenta uma maior taxa de satisfação em comparação com o [DSML4MAS](#)?
 - * **Métrica** - Questionário [SUS](#), número de comentários positivos e negativos ao longo da experiência.

- **Objetivo** - Avaliar a Aprendizagem do [SEA_ML++](#)
 - * **Pergunta:**
 1. **RQ5:** Depois de ter sido feita uma apresentação da linguagem, será que o desenvolvimento de sistemas com o [SEA_ML++](#) é uma tarefa complexa para principiantes em comparação com o [DSML4MAS](#)?
 - * **Métrica** - Tempo gasto em ambas as linguagens e dados obtidos pelo *Eye Tracker*.

- **Objetivo** - Avaliar a Memorização do [SEA_ML++](#)
 - * **Pergunta:**
 1. **RQ6:** Será que os símbolos do [SEA_ML++](#) são mais semanticamente transparentes em comparação com o [SEA_ML](#)?
 - * **Métrica** - Taxa de sucesso e coeficiente de transparência semântica.

- **Objetivo** - Avaliar a Taxa de Erros do [SEA_ML++](#)
 - * **Pergunta:**
 1. **RQ7:** Será que os participantes ao usarem o [SEA_ML++](#) cometem menos erros em comparação com o [DSML4MAS](#)?
 - * **Métrica** - Percentagem de erros cometidos ao longo da experiência.

5.2 Processo Experimental

Para efetuar a avaliação da usabilidade do [SEA_ML++](#), foi criado um processo Fig. 5.1 que consiste nas seguintes fases:

1. **Preparar Casos de Teste** - Determinar quais as tarefas que os utilizadores teste irão realizar.
2. **Sessão Piloto** - Esta sessão, deve ser feita por um utilizador que se encontre dentro dos utilizadores-alvo do sistema e permite garantir que os testes se encontram bem estruturados.
3. **Recrutar os Participantes** - Serão selecionados um conjunto de participantes experientes, com conhecimentos sobre o domínio do problema, e um outro de utilizadores inexperientes, sem esse tipo de conhecimentos.

4. **Sessão de Treino** - São apresentadas aos utilizadores ambas as linguagens, [SEA_ML++](#) e [DSML4MAS](#), com um exemplo. Este exemplo deve ser diferente dos casos de teste, de forma a garantir que os utilizadores não guardam em memória a resolução do problema.
5. **Explicar os Casos de Teste** - São explicados os exercícios que os participantes irão resolver. Nesta fase os participantes serão também informados que devem estar à vontade para fazer qualquer tipo de comentários, que não é o participante que está a ser testado, mas sim a linguagem, que os resultados individuais não serão publicados, mas sim os resultados gerais e que a anonimidade dos participantes será garantida. Finalmente, será apresentado um formulário de consentimento.
6. **Realizar Experiências de Usabilidade** - Nesta fase, as atividades dos participantes durante a realização dos exercícios são observadas e registadas. No fim, será feito um questionário aos participantes de forma a obter uma perspetiva qualitativa. Esta fase será explicada com mais detalhe no próximo sub-capítulo.
7. **Análise dos Dados** - São analisados cuidadosamente os dados de teste através do uso de determinadas métricas.

De seguida, caso necessário, é preciso aplicar algumas melhorias à linguagem [SEA_ML++](#), e voltar a repetir o processo com possivelmente mais utilizadores e com novos casos de teste garantindo assim que os exemplos resolvidos anteriormente não ficaram guardados na memória dos participantes.

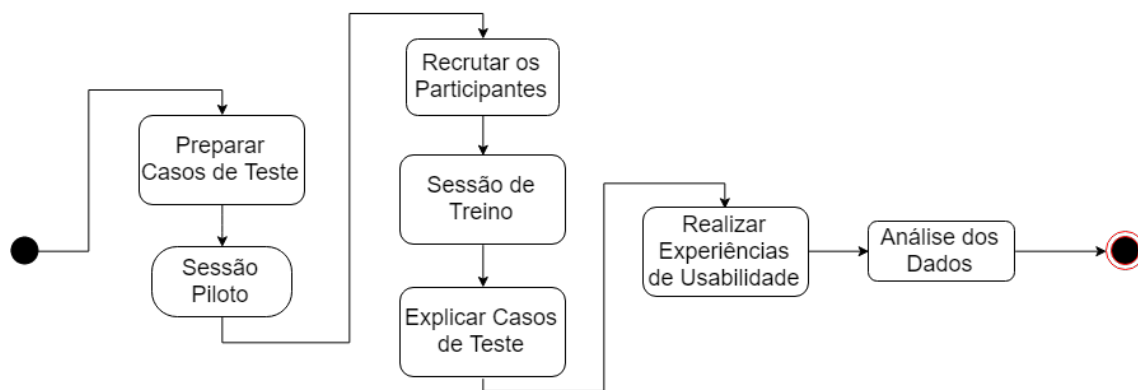


Figura 5.1: Processo de Avaliação Usabilidade.

AVALIAÇÃO TRANSPARÊNCIA SEMÂNTICA **SEA_ML** E **SEA_ML++**

6.1 Objetivos

SEA_ML é uma linguagem que apresenta ao utilizador um conjunto de construções semânticas distintas, estando estas divididas em 8 pontos de vista. Ao longo do tempo através de várias análises à sintaxe concreta do **SEA_ML**, e consequentemente tendo sido realizadas diversas melhorias, foi possível criar uma nova versão do **SEA_ML** denominada por **SEA_ML++**. De seguida, foram efetuadas algumas avaliações entre as notações originais e as notações propostas [Mir17]. Nestas avaliações não foi possível avaliar o coeficiente de transparência semântica, uma vez que não foram recolhidos dados de forma adequada que possibilitasse a realização de tal análise.

A transparência semântica permite avaliar a capacidade que os participantes têm para inferir os símbolos, tendo uma forma muito particular de ser medida. Para efetuar essa análise será replicado o desenho experimental usado em [Cai+13], onde foi desenvolvido um estudo semelhante focado na linguagem i^* , com algumas adaptações necessárias. Este método é muito usado para medir a compreensibilidade dos símbolos gráficos, sendo solicitado aos participantes que interpretem ou infiram o significado dos diversos símbolos existentes. No fim do processo, a compreensibilidade de um símbolo é normalmente medida pela percentagem de respostas corretas.

Assim, no presente estudo pretende-se avaliar a transparência semântica da nova linguagem, tendo sido este um dos fatores que ficou por desenvolver na primeira avaliação realizada à linguagem. Deste modo, será possível avaliar se de facto houve melhorias significativas na sua sintaxe concreta.

6.2 Material Experimental

O material experimental deste estudo encontra-se dividido em duas partes. A primeira parte consiste em dois questionários ambos com a mesma estrutura e cada um com 38 páginas, em que o objetivo é avaliar a transparência semântica de ambas as linguagens, [SEA_ML](#) e [SEA_ML++](#). Em cada página é apresentado, na parte superior, um símbolo seguindo de um conjunto de descrições em inglês de todos os símbolos a avaliar da linguagem. Assim, é pedido ao participante que faça corresponder a cada símbolo uma descrição que lhe parece mais correta. Tanto a ordem em que os símbolos são apresentados, como a ordem em que as descrições são listadas em cada página, irão ser alteradas de forma a neutralizar os efeitos de sequência.

Numa segunda parte é apresentado um questionário demográfico com 9 perguntas pessoais sobre o participante. Desta forma, serão recolhidos alguns dados para uma melhor caracterização dos participantes, como a idade, o género, a nacionalidade, se é estudante e/ou trabalhador, nestes casos é relevante questionar qual o curso que frequenta ou frequentou e/ou a função que desempenha no emprego atual. Para finalizar o questionário é perguntado aos participantes: 1. Experiência com Linguagens de Domínio Específico, 2. Experiência com Sistemas Multi Agente, 3. Experiência com Linguagens de Domínio Específico para Sistemas Multi Agente. Estas questões são de escolha múltipla existindo para cada uma 4 opções de resposta: 1. Nunca aprendi no contexto do curso, 2. Usei em contexto profissional, 3. Eu sei o que são mas nunca usei, 4. Nunca ouvir falar.

O questionário demográfico deste estudo encontra-se disponível no anexo [A](#).

6.3 Participantes

O processo experimental foi realizado por 24 participantes. Visto que todo o material deste estudo se encontra em inglês, para uma melhor replicação da experiência no Instituto Internacional de Informática da *Ege University*, um dos pré-requisitos na seleção do participante é terem um conhecimento básico da língua inglesa. De seguida são apresentados alguns dados sobre os participantes:

- A experiência foi realizada por 24 Portugueses, sendo 79,17% do sexo masculino e 20,83% do sexo feminino;
- 70,83% têm idades compreendidas entre os 22-26 anos, enquanto 29,17% têm idades compreendidas entre os 26-42 anos;
- 58,3% são estudantes, 25% são trabalhadores e 16,7% estudam e trabalham em simultâneo;
- Dos participantes estudantes, 82,4% são estudantes de Engenharia Informática enquanto 17,6% são estudantes de Informática e Gestão de Empresas;

- Em relação aos conhecimentos sobre [LDE](#), 12,5% participantes desenvolveram os seus conhecimentos no contexto do respetivo curso universitário. 12,5% reconhecem, porém nunca tiveram necessidade de as utilizar e 75% nunca ouviu falar neste tipo de linguagens;
- Em relação aos conhecimentos sobre [SMA](#), 70,83% sabem em que consiste mas nunca usaram e 29,17% nunca ouviram falar neste tipo de Sistemas;
- Em relação aos conhecimentos sobre [LDE](#) para [SMA](#), 100% dos participantes nunca ouviram falar neste tipo de linguagens.

6.4 Plano de Execução

O plano de execução (Fig. 6.1) está dividido nos seguintes passos:



Figura 6.1: Plano de Execução

1. **Questionário [SEA_ML](#)** - Os participantes foram informados para numa primeira instância lerem todas as descrições e posteriormente assinalar apenas uma descrição que consideram que melhor representa o símbolo apresentado no topo da página;
2. **Questionário [SEA_ML++](#)** - Estrutura e desenvolvimento semelhante ao questionário anterior, sendo que desta vez serão apresentadas as notações propostas por [\[Mir17\]](#);
3. **Questionário Demográfico** - Questionário onde os participantes respondem a um conjunto de informações para melhor caracterização do seu perfil.

Com o intuito de reduzir efeitos de aprendizagem, para além de ser trocada a ordem da linguagem com que os participantes respondiam ao questionário, foi também trocada a ordem das notações. 12 participantes começaram por responder ao questionário da linguagem [SEA_ML](#), e 12 participantes começaram pela linguagem [SEA_ML++](#).

Não foi estabelecido qualquer limite de tempo para a realização deste plano experimental, mas os participantes demoraram entre 15 a 22 minutos para o realizar.

6.5 Análise dos Resultados

Para realizar uma análise das respostas dos participantes optámos por recolher, manualmente as mesmas. Inicialmente foi feita uma comparação da percentagem de respostas

certas dadas por cada participante em cada linguagem e de seguida para cada símbolo foi calculada a taxa de sucesso no seu reconhecimento e o coeficiente de transparência semântica das duas sintaxes concretas, SEA_ML e SEA_ML++. As respostas dos participantes foram recolhidas para duas tabelas, uma para cada linguagem. Cada tabela possui 32 colunas, uma para cada sintaxe concreta, e 24 linhas, uma para cada participante. De seguida, é preenchido em cada coluna a resposta do participante para cada sintaxe concreta e caso o participante não tenha respondido, a célula é preenchida com o valor "NA" (do inglês "No Answer"). Desta forma, para além de ser possível contabilizar o número de respostas corretas por parte do participante, é também possível contabilizar o número de respostas alternativas.

Em relação à comparação dos resultados de cada participante, é possível verificar pela Fig. 6.2, que a grande maioria dos participantes respondeu corretamente a mais sintaxes concretas do SEA_ML++ em comparação com o SEA_ML. Mais especificamente, em média, os 24 participantes responderam corretamente a 41% das sintaxes concretas com o SEA_ML, que corresponde aproximadamente a 13 sintaxes e a 54% com o SEA_ML++ que corresponde aproximadamente a 17 sintaxes. De forma geral os participantes responderam a um maior número de respostas corretas com o SEA_ML++ em comparação com o SEA_ML, sendo este comportamento expectável uma vez que têm ocorrido melhorias ao longo do tempo na linguagem.

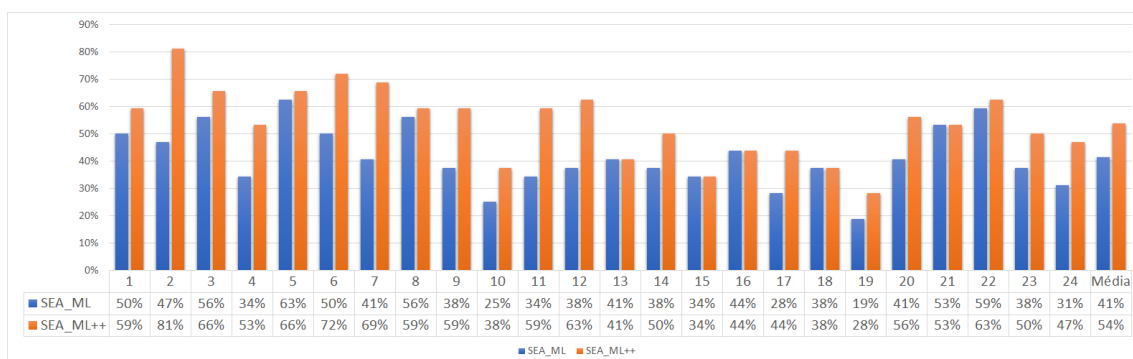


Figura 6.2: Percentagem Respostas Corretas dos Participantes

É necessário verificar também quais os símbolos que possuem uma baixa Taxa de Sucesso (do inglês *Hit Rate*). Pela Fig. 6.3, é possível verificar que as notações do SEA_ML++ são melhores em comparação com as do SEA_ML. Apesar disso, é possível verificar que apenas 14 dos 32 símbolos atingem o limite admissível para a compreensão de um símbolo, cuja a taxa de sucesso deve ser maior ou igual a 67% [Cai+13].

Destes 14 símbolos é possível concluir que:

- 8 dos símbolos já possuíam uma Taxa de Sucesso superior a 67% com a notação antiga. Ainda assim, houve uma melhoria significativa na Taxa de Sucesso com a nova notação tendo muitos atingido os 100%.

- 1 dos símbolos ("*Grounding*"), possui uma maior Taxa de Sucesso com a notação anterior do que com a nova notação. Apesar de na nova notação o valor não ser muito distante da anterior, encontra-se abaixo do valor mínimo recomendado;
- Nos restantes 5 símbolos houve uma grande melhoria na Taxa de Sucesso, tendo aumentado cerca de 39,2%.

Nos restantes 19 símbolos, que se encontram abaixo do valor admissível para a compreensão de símbolos, é possível verificar que:

- 4 símbolos possuem uma Taxa de Sucesso de 0% no [SEA_ML++](#);
- 7 símbolos possuem uma Taxa de Sucesso inferior a 10% no [SEA_ML++](#);
- 4 símbolos possuem uma Taxa de Sucesso superior a 60% mas inferior a 67% no [SEA_ML++](#);
- 9 símbolos possuem uma Taxa de Sucesso no [SEA_ML](#) superior ou igual à do [SEA_ML++](#), tendo dois deles uma taxa superior a 60%;

































































	SEA_ML	SEA_ML++		SEA_ML	SEA_ML++		SEA_ML	SEA_ML++
Action	 87,5 %	 91,7 %	Message	 100 %	 100 %	SSFP	 37,5 %	 75 %
Agent State	 16,7 %	 62,5 %	Message Sequence	 91,7 %	 95,8 %	SSRP	 41,7 %	 45,8 %
Agent Type	 4,2 %	 33,3 %	<u>ODMOWLClass</u>	 4,2 %	 4,2 %	Semantic Web Organization	 25,0 %	 70,8 %
Architecture Role	 12,5 %	 0 %	Ontology Role	 4,2 %	 4,2 %	Send	 95,8 %	 100 %
Behavior	 4,2 %	 62,5 %	Ontology Mediator Role	 4,2 %	 0 %	Service Ontology	 4,2 %	 0 %
Capability	 0 %	 41,7 %	Organization Ontology	 12,5 %	 20,8 %	Web Service	 95,8 %	 100 %
Domain Role	 4,2 %	 0 %	Plan	 12,5 %	 4,2 %			
Effect	 83,3 %	 87,5 %	Precondition	 62,5 %	 91,7 %			
Fact	 4,2 %	 41,7 %	Process	 62,5 %	 58,3 %			
Goal	 95,8 %	 100 %	Receive	 95,8 %	 100 %			
Grounding	 70,8 %	 66,7 %	Resource	 25 %	 75 %			
Interaction	 25 %	 37,5 %	SSAP	 37,5 %	 70,8 %			
Interface	 66,7 %	 16,7 %	SSEP	 37,5 %	 62,5 %			

Figura 6.3: Análise da Taxa de Sucesso (Verde = Acima do Limite; Sublinhado = Melhor)

Para uma distribuição destes dados, foram analisadas algumas medidas de estatística descritiva para facilitar a compreensão dos dados:

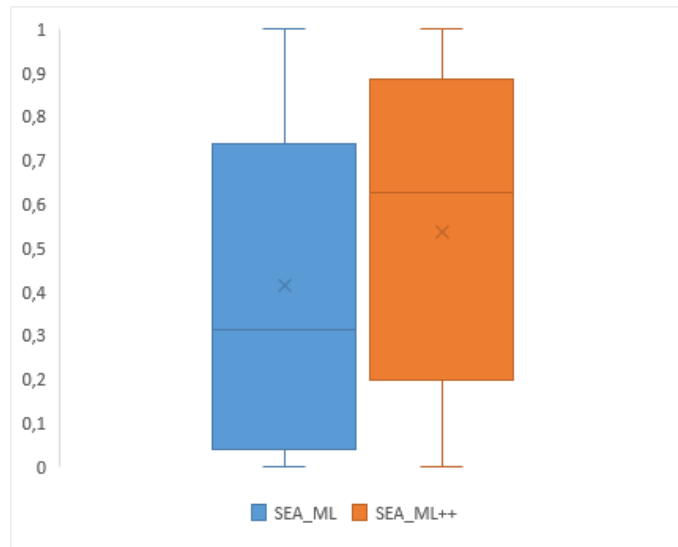
- Média: Valor que resume os dados de uma amostra;
- Mediana: Valor central que divide a metade menor da metade maior das observações;
- Moda: Permite analisar qual o valor mais repetido numa amostra;
- Desvio Padrão: Possibilita analisar qual o grau de variação da amostra;

- Assimetria (do inglês *Skew*): Permite analisar a forma das distribuições. Neste caso um valor negativo significa que os valores mais baixos das observações prevalecem, apresentando a função uma cauda mais longa do lado esquerdo. Caso a assimetria apresente um valor igual a 0 diz-se que a distribuição é simétrica, isto é, os valores estão distribuídos de forma idêntica em ambos os lados da média. Caso apresente um valor positivo significa que predominam os valores mais altos das observações, apresentando a função desta vez uma cauda mais longa do lado direito;
- Curtose (do inglês *Kurtosis*): Permite analisar o grau de achatamento da curva correspondente à distribuição. Caso apresente um valor negativo, a distribuição é mais achatada do que a distribuição normal (platicúrtica). Se a curtose apresentar um valor igual a 0, então a distribuição tem o mesmo achatamento da distribuição normal (mesocúrtica). Por fim, apresentando um valor positivo, a distribuição é mais pontiaguda em comparação com a distribuição normal (leptocúrtica);
- Teste Shapiro-Wilk (S-W): Possibilita analisar se uma determinada amostra possui uma distribuição normal. Caso o resultado S-W seja inferior ao valor α escolhido, então podemos afirmar que a distribuição não é normal ($S-W > \alpha$). Foi escolhido este teste, pois a amostra possui uma dimensão igual a 32 símbolos, e de acordo com [RW+11] é um teste ideal para dimensões inferiores a 50 elementos.

Em relação à média da Taxa de Sucesso é possível verificar pela Tabela 6.4(b), que o [SEA_ML++](#) e o [SEA_ML](#) apresentam, em média, taxas de sucesso de 53,8% e 41,4%, respectivamente. Estes valores evidenciam numa diferença de 12,4% representando uma melhoria da primeira linguagem para a segunda. Essa melhoria pode também ser verificada através do *boxplot* 6.4(a). Ainda assim, a média da Taxa de Sucesso do [SEA_ML++](#) encontra-se muito abaixo do valor admissível para a compreensão de símbolos gráficos, 67%. Adicionalmente verifica-se que a mediana apresenta o valor de 31,3% no [SEA_ML](#) e de 62,5% no [SEA_ML++](#).

A Taxa de Sucesso de 4,2% no [SEA_ML](#) é a mais comum, sendo que no [SEA_ML++](#) verificamos que a moda é de 100%. Aparentemente estaríamos perante um sinal positivo tendo em conta que 5 símbolos possuem uma Taxa de Sucesso de 100%, porém ao olharmos para a média, é de apenas 53,8%. Esta discrepância ocorre uma vez que 3 símbolos desta última linguagem apresentam sucesso nulo, isto é, Taxa de Sucesso de 0%. Deste modo existem símbolos que mostram ter uma elevada compreensibilidade, enquanto outros encontram-se no extremo oposto. Assim, é possível concluir que a moda não é a ferramenta mais relevante para este estudo.

O desvio padrão apresenta um valor bastante idêntico em ambas as versões, 35%, sendo a variação dos dados da amostra aproximadamente igual. Estes valores são elevados e corroboram com a conclusão abaixo desenvolvida, quando analisados os valores da curtose.



((a)) Comparação entre as Taxas de Sucesso

	<i>SEA_ML</i>	<i>SEA_ML++</i>
Média	41,40%	53,80%
Mediana	31,30%	62,50%
Moda	4,20%	100%
Desvio Padrão	35,30%	35,60%
Assimetria	0,42	-0,24
Curtose	-1,43	-1,35
Shapiro-Wilk	0,000312	0.003043

((b)) Estatística descritiva sobre os resultados da Taxa de Sucesso

Figura 6.4: Resultados sobre a Taxa de Sucesso

A assimetria no *SEA_ML* é de 0,42, isto significa que a distribuição é assimétrica. Assim, é possível concluir que a cauda mais longa da distribuição encontra-se à direita, indicando menor a ocorrência de valores mais altos. No sentido contrário, a assimetria no *SEA_ML++* é de -0,24, neste caso a cauda mais longa encontra-se lado esquerdo da distribuição, mostrando existir uma maior concentração de dados nos valores mais altos. Podemos suportar esta conclusão através da comparação entre a média, moda e mediana. Em casos em que a assimetria é negativa, a média é inferior à mediana que, por sua vez, é inferior à moda, como mostra a Tabela 6.4(b) em relação ao *SEA_ML++*. Quando a média mostra ser o valor maior, seguido da mediana e da moda estamos perante uma assimetria positiva, que é o caso do *SEA_ML*.

Em relação à curtose, tanto o *SEA_ML* como o *SEA_ML++* apresentam valores negativos, -1,43 e -1,35 respetivamente. Estes valores indicam que os dados se encontram muito distribuídos, isto é, em ambos os casos não existe uma grande concentração dos dados.

Em relação ao teste de Shapiro-Wilk, este foi realizado com um valor *alpha* igual a 0,05. Em ambos os casos o valor S-W apresentam valores < 0,05, deste modo podemos concluir que as amostras não apresentam uma distribuição normal.

De acordo com [Cai+13], o problema com as medidas convencionais para a compreensão de símbolos, como a Taxa de Sucesso, é que os resultados não podem ter valores negativos. Assim, seguindo o estudo de [Cai+13], será necessário analisar o coeficiente de Transparência Semântica, pois através deste é possível avaliar a transparência semântica numa escala de -1 a +1. Quando obtido um valor positivo o símbolo é transparente (o símbolo sugere o seu significado) e ao ser negativo indica que a sua aparência implica um significado incorreto (semanticamente perverso).

A tabela 6.5 apresenta os resultados do coeficiente de Transparência Semântica de cada elemento da sintaxe concreta de cada uma das linguagens. Na primeira coluna da tabela é apresentado o nome do símbolo e as duas colunas seguintes contêm os valores do coeficiente de Transparência Semântica tanto para o SEA_ML como para o SEA_ML++. Foi também sublinhado o valor do coeficiente mais elevado entre as duas sintaxes concretas, ou seja, o símbolo mais semanticamente transparente. Complementarmente, foram aplicadas três cores nas células da tabela 6.5 que representa o grau de Transparência Semântica dos símbolos. As células com um fundo a verde são símbolos que são semanticamente transparentes, as a amarelo têm um grau inferior de Transparência Semântica mas positivo, e as células a vermelho têm um grau baixo de Transparência Semântica.

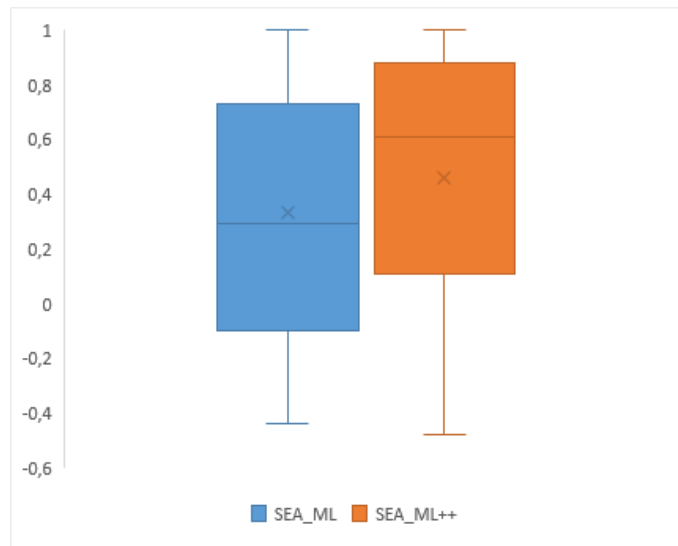
	SEA_ML	SEA_ML++
Action	0,87	0,91
AgentState	-0,27	0,61
AgentType	-0,10	0,31
ArchitectureRole	0,10	-0,10
Behaviour	-0,14	0,61
Capabilities	-0,44	0,40
DomainRole	-0,05	-0,40
Effect	0,83	0,87
Fact	-0,1	0,40
Goal	1,0	1,00
Grounding	0,70	0,66
Interaction	0,23	-0,48
Interface	0,66	0,14
Message	1,00	1,00
MessageSequence	0,91	0,96
ODMOWLClass	-0,10	-0,35
OntologyRole	-0,05	-0,14
OntologyMediatorRole	-0,18	-0,27
OrganizationOntology	-0,10	0,18
Plan	0,10	0,01
Precondition	0,61	0,91
Process	0,61	0,57
Receive	0,96	1,00
Resource	0,23	0,74
SSAP	0,35	0,70
SSEP	0,35	0,61
SSFP	0,35	0,74
SSRP	0,40	0,44
SemanticWebOrganization	0,23	0,70
Send	0,96	1,00
ServiceOntology	-0,14	-0,05
WebService	0,96	1,00

Figura 6.5: Coeficiente de Transparência Semântica

Ainda sobre uma visão geral da tabela 6.5, é possível verificar que houve uma melhoria significativa do coeficiente de Transparência Semântica na maioria dos símbolos. Ainda

assim, é de salientar que ainda existem 10 símbolos com um coeficiente de Transparência Semântica muito baixo, tendo 7 deles um coeficiente negativo.

Pela tabela 6.6(b), é possível verificar que a média no **SEA_ML++** é superior à média do **SEA_ML**, tal como verificado através da análise anterior. As médias assumem os valores de 0,46 e 0,33, respetivamente, o que representa um aumento no coeficiente de Transparência Semântica de 0,13. Esta melhoria também pode ser observada no *boxplot* 6.6(a).



((a)) Comparação entre os Coeficientes de Transparência Semântica

	SEA_ML	SEA_ML++
Média	0,33	0,46
Mediana	0,29	0,61
Moda	-0,1	1
Desvio Padrão	0,45	0,47
Assimetria	0,1423	-0,5642
Curtose	-1,3596	-0,8896
Shapiro-Wilk (S-W)	0,0134	0,0061

((b)) Estatística descritiva sobre os Coeficientes de Transparência Semântica

Figura 6.6: Resultados sobre o Coeficiente de Transparência Semântica

Em relação à mediana em ambas as linguagens o valor é positivo, sendo que no **SEA_ML++** esta medida estatística assume o valor de 0,61 enquanto no **SEA_ML** o valor central é 0,29.

O valor da moda no **SEA_ML** é um valor negativo, -0,1, porém no **SEA_ML++** assume o máximo de compreensibilidade, valor 1. Conclusão idêntica à obtida anteriormente, quando através da medida convencional o valor mais comum foi de 100%. A média desta última linguagem é influenciada, em parte, pelo mínimo de coeficiente de transparência semântica ser de -0,48, enquanto no **SEA_ML** o mínimo é de -0,44. Por outro lado, o máximo de coeficiente de transparência semântica é 1 em ambos os casos.

O desvio padrão é idêntico em ambas as linguagens, assumindo apenas 0,02 unidades

de diferença entre o [SEA_ML](#) e o [SEA_ML++](#) (0,45 e 0,47, respetivamente).

A assimetria, tal como na Taxa de Sucesso, apresenta um valor positivo no [SEA_ML](#) (0,1423), havendo assim poucos valores altos - cauda mais longa à direita -, ao contrário do [SEA_ML++](#) (-0,5642) onde há um maior número de valores altos - cauda mais longa à esquerda.

O valor curtose apresentam ambos valores negativos, mas neste caso como a diferença dos valores é maior em comparação à Taxa de Sucesso, é possível afirma que existe uma maior concentração de dados no [SEA_ML++](#) (pouca).

Por último em relação ao teste Shapiro-Wilk, foi realizado com um valor *alpha* igual a 0,05. Tal como na Taxa de Sucesso, em ambos os casos, os valores S-W mostram ser inferiores a 0,05. Deste modo é possível concluir que ambas não apresentam uma Distribuição Normal.

6.6 Teste de Hipótese

RQ6: Será que os símbolos do [SEA_ML++](#) são mais semanticamente transparentes em comparação com o [SEA_ML](#) ?

Depois de feita uma análise à Taxa de Sucesso e ao Coeficiente de Transparência Semântica, é possível afirmar que em geral, os símbolos do [SEA_ML++](#) apresentam uma melhor performance. Esta conclusão é suportada pelo *Wilcoxon Test* que possibilita comparar duas amostras relacionadas de distribuição não normal. Através da tabela 6.1, podemos verificar que para a Taxa de Sucesso o [SEA_ML++](#) apresenta um maior número de classificações positivas (21) em comparação com o [SEA_ML](#) (8). Adicionalmente, dos 32 símbolos apenas três apresentam valores iguais. Do mesmo modo, para o Coeficiente de Transparência Semântica, pela tabela 6.2 podemos validar que para o [SEA_ML++](#) existe um maior número de classificações positivas (20) em comparação com o [SEA_ML](#) (10). Neste caso, dos 32 símbolos apenas dois apresentam valores iguais.

		N	Pontos de Média	Soma de Classificações
SEA_ML - SEA_ML ++	Classificações Negativas	8 ^a	11,81	94
	Classificações Positivas	21 ^b	16,21	340,50
	Vínculos	3 ^c		
	Total	32		

Tabela 6.1: Wilcoxon Test para a Taxa de Sucesso (a-[SEA_ML++](#)<[SEA_ML](#) , b-[SEA_ML++](#)>[SEA_ML](#) , c-[SEA_ML++](#)=[SEA_ML](#))

Isto significa que, em média, os símbolos [SEA_ML++](#) assumem valores superiores de Taxa de Sucesso assim como, maiores coeficientes de transparência semântica, quando comparados com os resultados do [SEA_ML](#). Através dos resultados inerente à Taxa de Sucesso podemos concluir que os 5 símbolos com maior percentagem são iguais em ambas as linguagens - Message, Goal, Receive, Send e Web Service. Sendo que no [SEA_ML++](#)

		N	Pontos de Média	Soma de Classificações
SEA_ML - SEA_ML ++	Classificações Negativas	10 ^a	14,50	145
	Classificações Positivas	20 ^b	16	320
	Vínculos	2 ^c		
	Total	32		

Tabela 6.2: Wilcoxon Test para o Coeficiente de Transparência Semântica
(a-SEA_ML++<SEA_ML , b-SEA_ML++>SEA_ML , c-SEA_ML++=SEA_ML)

todos estes símbolos apresentam uma Taxa de Sucesso de 100%, no SEA_ML apenas o primeiro símbolo referido apresenta esta percentagem, tendo os restantes uma Taxa de Sucesso de 95,8%.

Na análise ao Coeficiente de Transparência Semântica ao nível do SEA_ML, num total de 32 símbolos, 21 apresentam um coeficiente superior a 0. Um coeficiente com valor positivo significa que é semanticamente transparente. Destes símbolos com coeficiente positivo, 5 apresentam um coeficiente de transparência semântica maior que 0 mas menor que 0,20, denominados por semanticamente translúcidos. Os restantes 11 símbolos apresentam um coeficiente de transparência semântica inferior a 0, sendo considerados semanticamente perversos.

Ao nível do SEA_ML++ o número de símbolos semanticamente transparente é superior (25 símbolos), sendo apenas 3 destes semanticamente translúcidos. Coeficientes inferiores a 0 nesta linguagem existem somente 7, ou seja, menos coeficientes semanticamente perversos. No geral esta linguagem apresenta um nível de compreensibilidade superior, esta conclusão é suportada tanto pela análise à Taxa de Sucesso, assim como pela análise aos Coeficientes de Transparência Semântica.

Seria interessante replicar esta experiência com utilizadores com conhecimentos sobre Linguagens de Domínio Específico para Sistemas Multi Agentes. Através destes utilizadores seria possível verificar se os resultados se mantêm ou se existe melhoria dos dados obtidos.

6.7 Ameaças à validade do estudo

Neste estudo esteve envolvido um grupo de 24 participantes, sendo que cada um realizou o questionário em ambas as linguagens, SEA_ML e SEA_ML++, o que dá um total de 42 questionários para analisar. O número de participantes e o método de seleção utilizado apresentam um risco, uma vez que os resultados obtidos podem não ser iguais quando usada uma amostra com participantes de diferentes tipos.

Outra limitação desta avaliação, é o facto de existirem poucos participantes experientes. Apesar de um dos objetivos ser tornar a linguagem mais acessível a utilizadores inexperientes, era também um ponto de interesse realizar este estudo com alunos que tivessem formação em SMA e neste tipo de linguagens. Devido à dificuldade de coordenação com a

agenda letiva, tornou a amostra um pouco mais limitada nesta dissertação. Porém, esta limitação tornar-se-á numa investigação futura, uma vez que estão previstas replicações deste estudo com utilizadores experientes na universidade de *Ege*.

AVALIAR A **SEA_ML++** COMPARANDO-A COM O **DSML4MAS**

7.1 Objetivos

O objetivo principal desta experiência é avaliar a linguagem **SEA_ML++** em relação à sua sintaxe abstrata. Para tal será feita uma comparação com uma outra linguagem de modelação de **SMA**, **DSML4MAS**. Cada participante irá executar duas tarefas de modelação: uma usando o **SEA_ML++**, e outra usando o **DSML4MAS**. No **SEA_ML++** o utilizador irá modelar os pontos de vista *Agent*, *MasAndOrg* e *SWSInteraction*, enquanto no **DSML4MAS** o utilizador irá modelar os pontos de vista *Agent*, *Mas* e o *Organization*. Foram escolhidos estes pontos de vista, pois para além de existir uma relação direta entre cada ponto de vista de cada linguagem, é também possível que qualquer participante não experiente consiga entender a maioria dos conceitos presentes nos pontos de vista.

O objetivo principal deste estudo é analisar o efeito do uso do **SEA_ML++** e **DSML4MAS**, para fins de avaliação em relação à eficácia, eficiência, satisfação, aprendizagem e taxa de erros com que um participante modela um **SMA**.

7.2 Tarefas

Cada participante será solicitado para executar duas tarefas de modelação: uma usando **SEA_ML++** e outra usando **DSML4MAS**. As duas tarefas têm uma complexidade semelhante e consistem na modelação de um **SMA**, utilizando o editor *Eclipse*, a partir de uma descrição textual. Este editor, apenas varia nas construções de linguagem e nas regras de composição oferecidas aos participantes, dependendo da linguagem que o participante está a usar. Independentemente da tarefa a desenvolver, o participante tem a tela do ecrã dividida, com a maior parte sendo ocupada pelo editor, no lado esquerdo, e uma parte

menor com o caso de estudo que o participante deve modelar, no lado direito. A Fig. 7.1 apresenta um exemplo do ponto de partida para executar uma tarefa com o [SEA_ML++](#). Tanto a descrição textual do modelo, no lado direito, quanto o editor, à esquerda, são dimensionados de modo a que todo o exercício possa ser executado sem a necessidade de redimensionar qualquer janela. Adicionalmente, os participantes são instruídos a não alterar os tamanhos das janelas de forma a possibilitar a comparabilidade entre as sessões. Depois de executar as duas tarefas de modelação, os participantes são solicitados a responder a um questionário [SUS](#) sobre o [SEA_ML++](#) e [DSML4MAS](#).

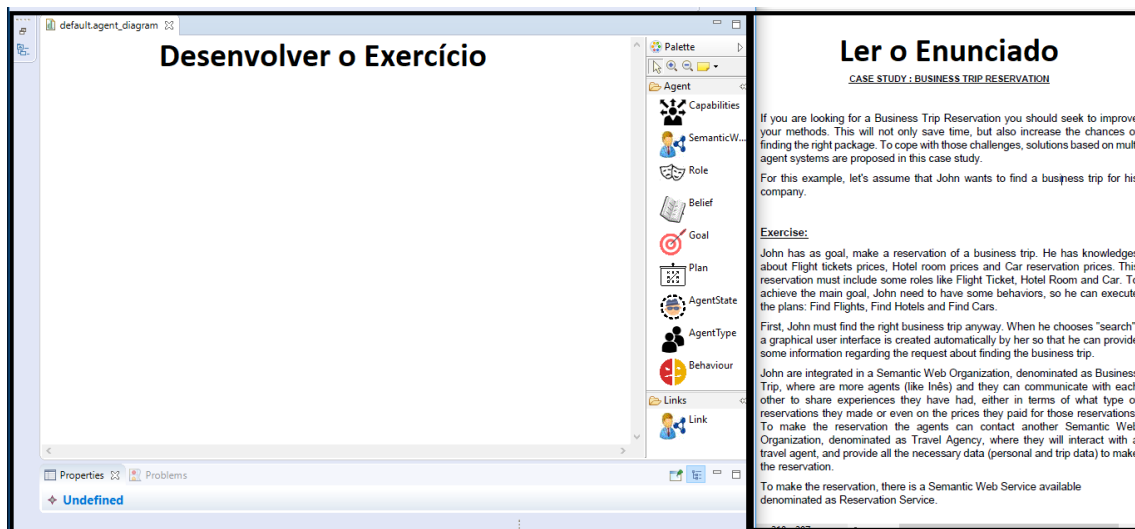


Figura 7.1: Tela Monitor

A Tabela 7.2, descreve o desenho experimental escolhido. Foram utilizados dois desafios diferentes (D1 e D2), com complexidades semelhantes. Cada participante resolveu estes dois desafios usando uma linguagem em cada um deles. Para reduzir os efeitos de aprendizagem, foram utilizados grupos de 4 utilizadores, sendo que cada participante pertencia a um grupo A,B,C ou D.

Gr.	Carta	Quest.	Tut.	Cal.	Desafio 1	Desafio 2	SUS
A	✓	✓	✓	✓	D1 / SEA_ML++	D2 / DSML4MAS	✓
B	✓	✓	✓	✓	D1 / DSML4MAS	D2 / SEA_ML++	✓
C	✓	✓	✓	✓	D2 / SEA_ML++	D1 / DSML4MAS	✓
D	✓	✓	✓	✓	D2 / DSML4MAS	D1 / SEA_ML++	✓

Figura 7.2: Desenho Experimental e Sequência de Tarefas

7.3 Material Experimental

O material experimental desta experiência inclui:

- **Carta de Consentimento** - O utilizador irá ler e assinar uma carta de consentimento (anexo B);
- **Vídeo de Apresentação** - O utilizador irá assistir a um vídeo de apresentação de ambas as linguagens (SEA_ML++ e DSML4MAS), de forma a que sejam assimilados alguns conceitos relevantes para o desenvolvimento do caso de teste;
- **Monitor/Rato/Teclado** - O utilizador irá resolver um caso de teste de modelação;
- **Questionário de Satisfação** - Permite quantificar o grau de satisfação dos participantes após desenvolverem o caso de teste (anexo C).

7.4 Participantes

Visto que todos os participantes realizaram ambas as experiências e com a ajuda do questionário demográfico obtido na experiência anterior (6), foi possível concluir que:

- 79% dos participantes são do sexo Masculino e 21% são do sexo Feminino (Fig. 7.3(a));
- A média de idade dos participantes é 27 anos (Fig. 7.3(b));
- 75% dos participantes são estudantes sendo 83% estudantes de Engenharia Informática e 17% estudantes de Informática e Gestão de Empresas. Incorporado neste grupo encontram-se 4 participantes que adicionalmente já se encontram a desempenhar funções numa empresa na área. Da amostra total os restantes 25% são exclusivamente trabalhadores (Fig. 7.3(c));
- Em relação aos conhecimentos sobre LDE, 75% dos participantes nunca ouviram falar, 12,5% sabem o que são mas nunca usaram e 12,5% aprenderam no contexto do curso. Os conhecimentos sobre SMA, 29% dos participantes não têm qualquer conhecimento e 71% conhecem, porém nunca utilizaram. Finalmente, os conhecimentos sobre LDE para SMA, 100% dos participantes nunca ouviram falar (Fig. 7.3(d)).

7.5 Caso de estudo

Nesta experiência foram criados dois casos de teste, também em inglês para uma melhor replicação da experiência no Instituto Internacional de Informática da *Ege Univeristy*, denominado de *Business Trip Reservation* - Reserva de uma Viagem Empresarial e *Restocking Warehouse* - Reposição de Armazém.

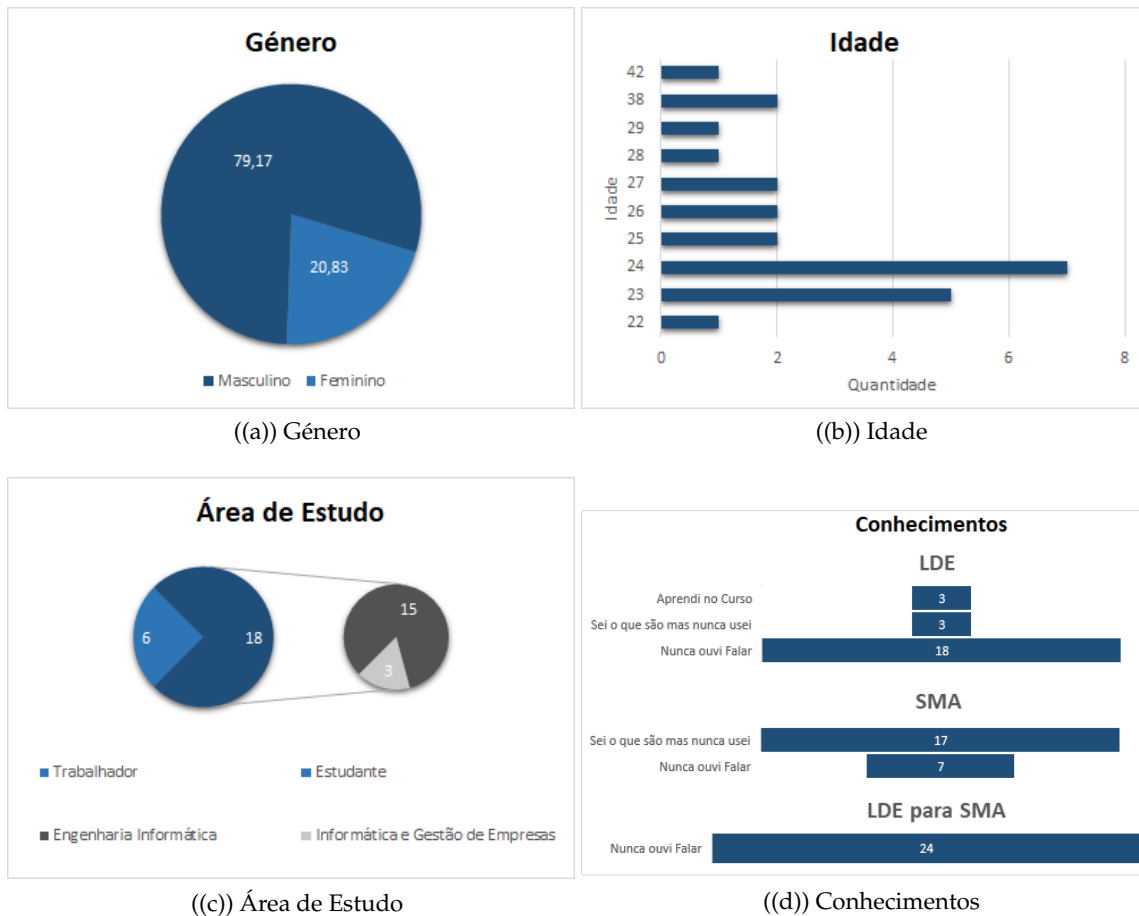


Figura 7.3: Resultados Questionário Demográfico

7.5.1 Reserva de uma Viagem Empresarial

Neste caso de estudo, era solicitado aos participantes que desenvolvessem um sistema que permitisse a um agente efetuar uma reserva de viagem de negócios, sendo que essa viagem deve incluir voo, hotel e carro de aluguer. O enunciado contém todas as variáveis do sistema que devem ser modeladas. Para cada linguagem, em cada ponto de vista, é necessário que o utilizador modele alguns conceitos necessários para o funcionamento do sistema. A descrição completa do caso de estudo encontra-se disponível no anexo D.

7.5.1.1 Solução Possível

O agente *John* tem uma capacidade denominada por *Trip Reservation* que inclui o seu objetivo *Find Business Travel*, as suas crenças *Flight tickets Price Knowledge*, *Hotel Room Price Knowledge*, *Car Reservation Price Knowledge* e os seus planos *Find Flight*, *Find Hotel* e *Find Car*. Ao considerar as crenças, o agente irá precisar delas para decidir se está a fazer uma boa opção quando for reservar quer um voo, quer um quarto de hotel ou um carro. A função *Find Flight Ticket* poderá realizar a sua tarefa sobre o comportamento *Finding Flight Ticket* chamando o plano *Find Flight*. Do mesmo modo, com a função *Find Hotel Room*, o

agente pode adotar o comportamento *Finding Hotel Room* chamando o plano *Find Hotel* e a função *Find Car Available* pode ser também chamada adotando o comportamento *Find Car Available* que por sua vez irá chamar o plano *Find Car*. Finalmente, este ponto de vista inclui ainda algumas informações sobre o próprio agente, como o tipo e o estado do agente. Neste caso, o tipo do agente é *Business Trips* e é iniciado com um estado ativo.

A Fig. 7.4, mostra uma solução possível na linguagem *SEA_ML++*, enquanto a Fig. 7.5, mostra uma solução possível na linguagem *DSML4MAS*.

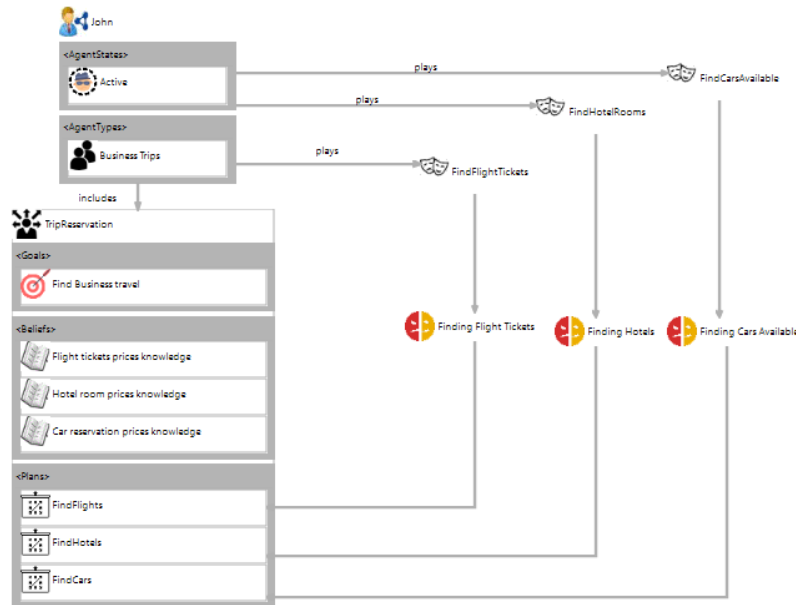


Figura 7.4: *SEA_ML++* ponto de vista *Agent*

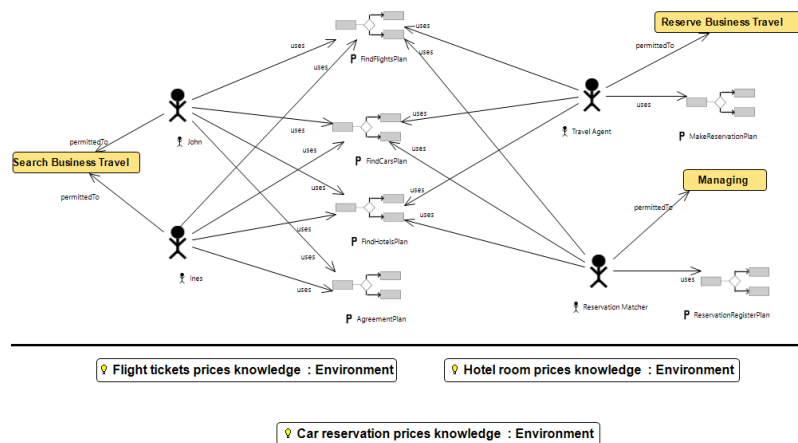


Figura 7.5: *DSML4MAS* ponto de vista *Agent*

Ao considerar a estrutura do sistema, os agentes da web semântica trabalham numa organização denominada por *Business Trips*, onde residem os agentes. Numa outra organização denominada por *Travel Agency* reside o agente *Travel Agent* e na organização *Management* reside o agente *Reservation Matcher* (uma instância do agente *SSMatchmaker*).

A organização *Business Trips* tem como função *Search Business Travel*, a organização *Travel Agency* tem como função *Reserve Business Travel* e por último a organização *Management* tem como função *Managing*. Todas estas organizações necessitam de aceder a alguns recursos que se encontram noutros ambientes. Assim, todas as organizações interagem com o ambiente *Reservation*, necessário para obter recursos e permissões de acesso. Todos os agentes interagem com o agente *Reservation Matcher* para aceder as reservas. Para além disso, todos os agentes interagem com o agente *Travel Agent* para fornecer todos os dados necessários para efetuar uma reserva. Por último, os agentes *John* e *Inês* interagem um com o outro para partilhar experiências que já tenham tido, seja em termos de que tipo de reservas fizeram ou até mesmo sobre os preços que pagaram por essas reservas.

A Fig. 7.6, mostra uma solução possível na linguagem [SEA_ML++](#), enquanto a Fig. 7.7, mostra uma solução possível na linguagem [DSML4MAS](#).

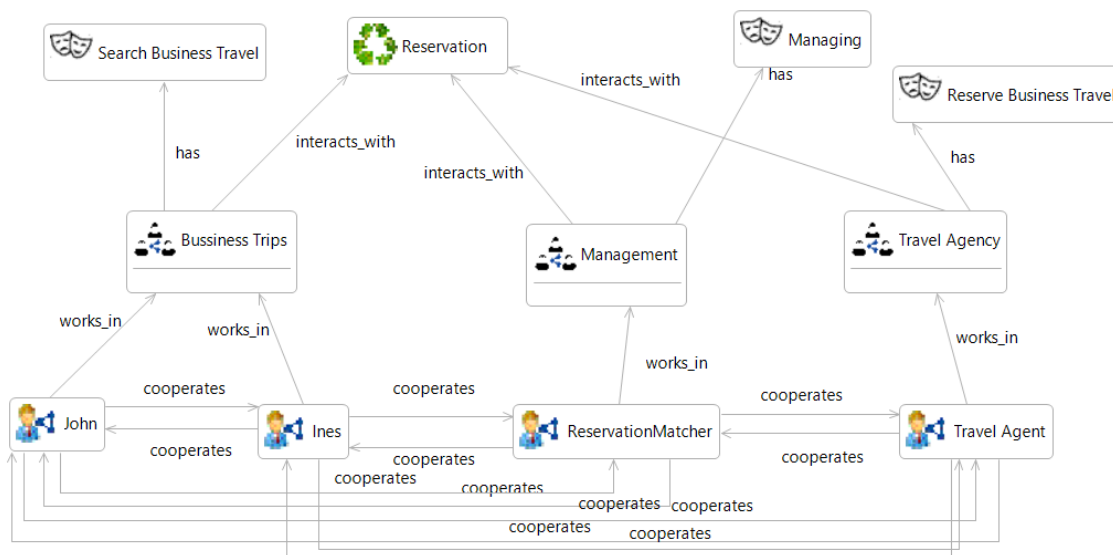


Figura 7.6: [SEA_ML++](#) ponto de vista *MasAndOrg*

É também relevante modelar os serviços e planos necessários para que o sistema funcione corretamente. Os agentes *John* e *Travel Agent* podem ser modelados para encontrar, *SS_FinderPlan*, fazer um acordo, *SS_AgreementPlan* e executar um serviço, *SS_ExecutorPlan*. Os serviços podem também ser modelados para uma interação entre os *Semantic Web Services* e os planos. Neste caso, quando é efetuada uma solicitação para encontrar uma viagem de negócios, o agente executaria a função de procurar uma viagem de negócios que aplicaria o plano *FindFlights*, *FindHotels* e *FindCars* para encontrar uma interface do serviço de reserva apropriada de um *SWS ReservationService*.

Esses planos realizam a sua descoberta interagindo com o agente *Reservation Matcher* que registou os serviços aplicando o plano *Reservation Register*. O agente *John* pode interagir com o *Travel Agent* para receber alguns conselhos de especialistas para a sua reserva, e de seguida o agente *Travel Agent* aplica o seu plano de *Agreement* para negociar com os serviços já descobertos. Essa negociação seria feita através da interface *ReservationInfo*

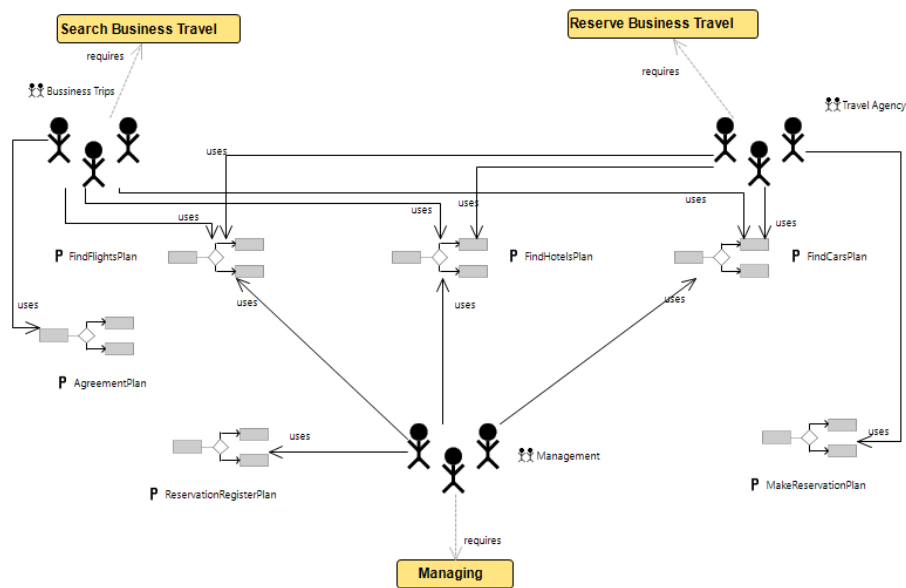


Figura 7.7: DSML4MAS ponto de vista *Organization*

do SWS *Reservation Service*. Por último, caso o resultado das negociações for positivo, o agente aplicaria o plano *Make Reservation* para chamar o *Reservation* do SWS executando o processo *Reservation Process* usando o *ReservationInfoAccess*.

A Fig. 7.8, mostra uma solução possível na linguagem *SEA_ML++*, enquanto a Fig. 7.9, mostra uma solução possível na linguagem *DSML4MAS*.

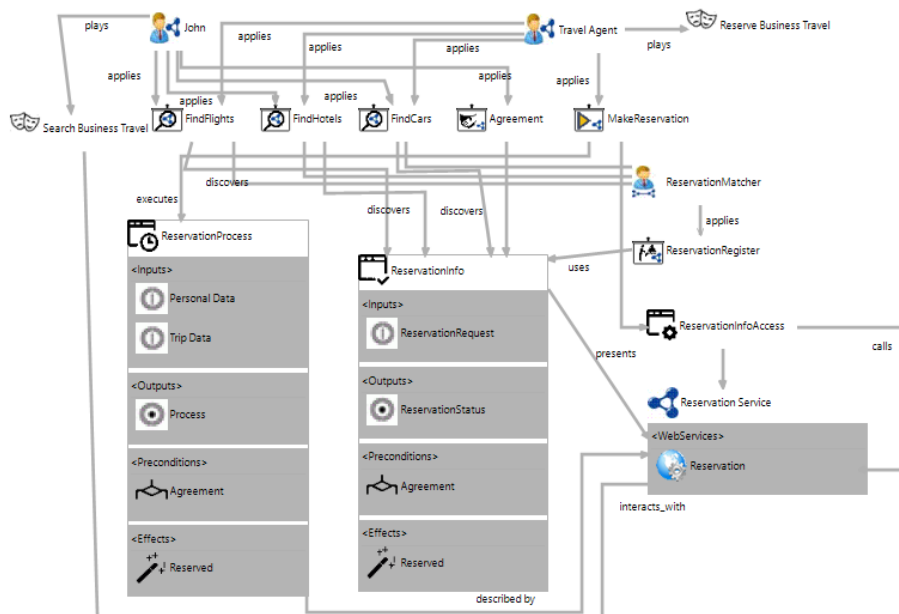
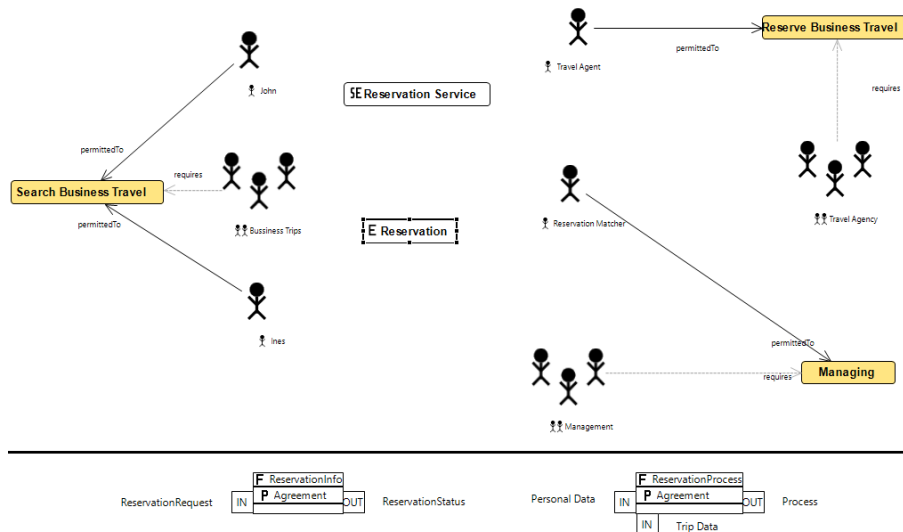


Figura 7.8: *SEA_ML++* ponto de vista *SWSInteraction*

Figura 7.9: [DSML4MAS](#) ponto de vista *Mas*

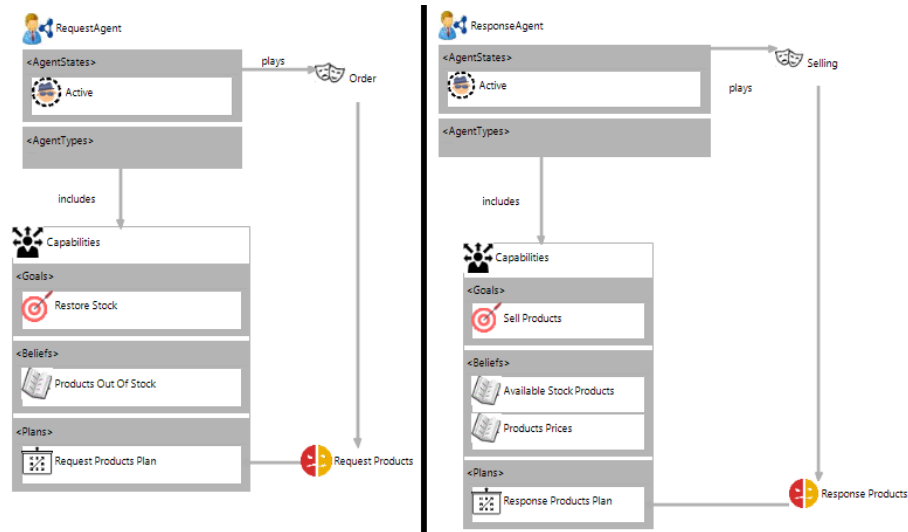
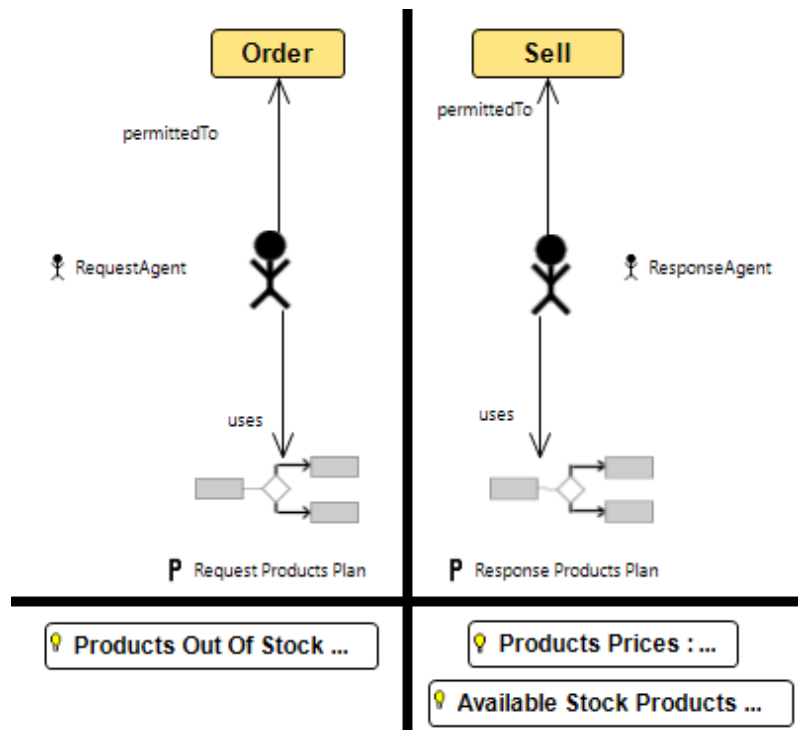
7.5.2 Reposição de Armazém

Neste caso de estudo, solicita-se aos participantes que desenvolvam um sistema que permita repor o seu stock de armazém automaticamente. O enunciado contém todas as variáveis do sistema que devem ser modeladas. Para cada linguagem, em cada ponto de vista, é necessário que o utilizador modele alguns conceitos necessários para o funcionamento do sistema. A descrição completa do caso de estudo encontra-se disponível no anexo [E](#).

7.5.2.1 Solução Possível

O agente *Request Agent*, encontra-se no estado Ativo. Este agente tem uma capacidade denominada por *Capabilities* que inclui o seu objetivo *Restore Stock* a sua crença *Products Out of Stock* e o seu plano *Request Products Plan*. Ao considerar a sua crença, o agente irá precisar dela para decidir se deve encomendar novos produtos ou não. Caso o agente opte por efetuar uma reposição do stock, a função *Order* é chamada adotando o comportamento *RequestProducts* que, por sua vez, irá chamar o plano *Request Products Plan*. O agente *Response Agent*, encontra-se no estado Ativo e tem uma capacidade denominada por *Capabilities* que inclui o seu objetivo *Sell Products*, as suas crenças *Available Stock Products* e *Products Available*, e o seu plano *Response Products Plan*. Ao considerar a sua crença, o agente irá precisar dela para decidir se deve vender os seus produtos ou não, caso disponíveis. Caso o agente opte por efetuar uma venda, a função *Sell* é chamada adotando o comportamento *ResponseProducts* que, por sua vez, irá chamar o plano *Response Products Plan*.

A Fig. [7.10](#), mostra uma solução possível na linguagem [SEA_ML++](#) enquanto a Fig. [7.11](#), mostra uma solução possível na linguagem [DSML4MAS](#).

Figura 7.10: SEA_ML++ ponto de vista *Agent*Figura 7.11: DSML4MAS ponto de vista *Agent*

Os agentes da Web Semântica trabalham numa organização denominada por *RequestResponseOrganization*, onde residem os agentes *Request Agent* e *Response Agent*. Esta organização tem como função *Order* e *Sell*, e interagem num ambiente *Communication*, necessário para obter recursos e permissões de acesso.

A Fig. 7.12, mostra uma solução possível na linguagem [SEA_ML++](#) enquanto a (Fig. 7.13), mostra uma solução possível na linguagem [DSML4MAS](#).

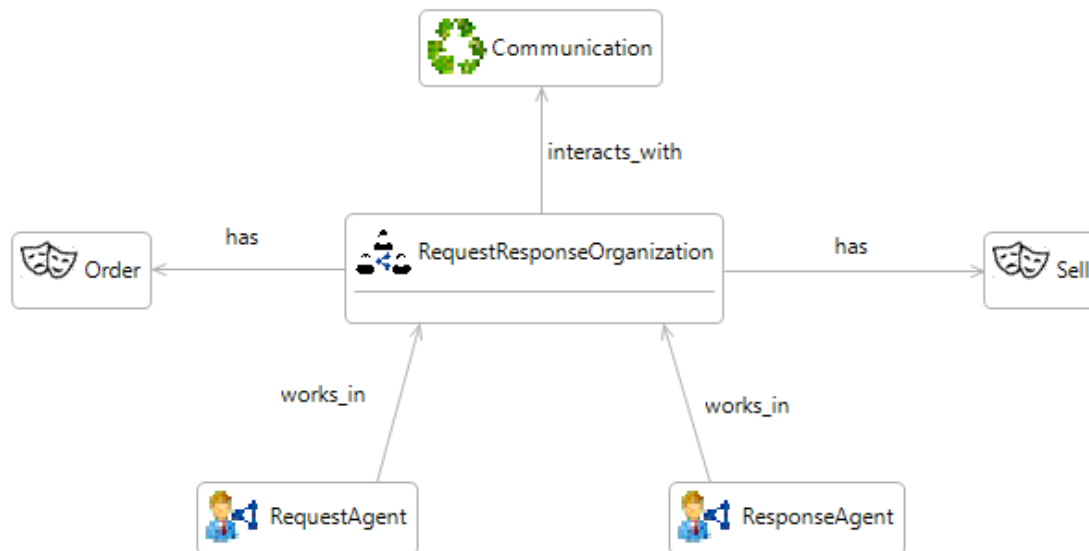


Figura 7.12: [SEA_ML++](#) ponto de vista *MasandOrg*

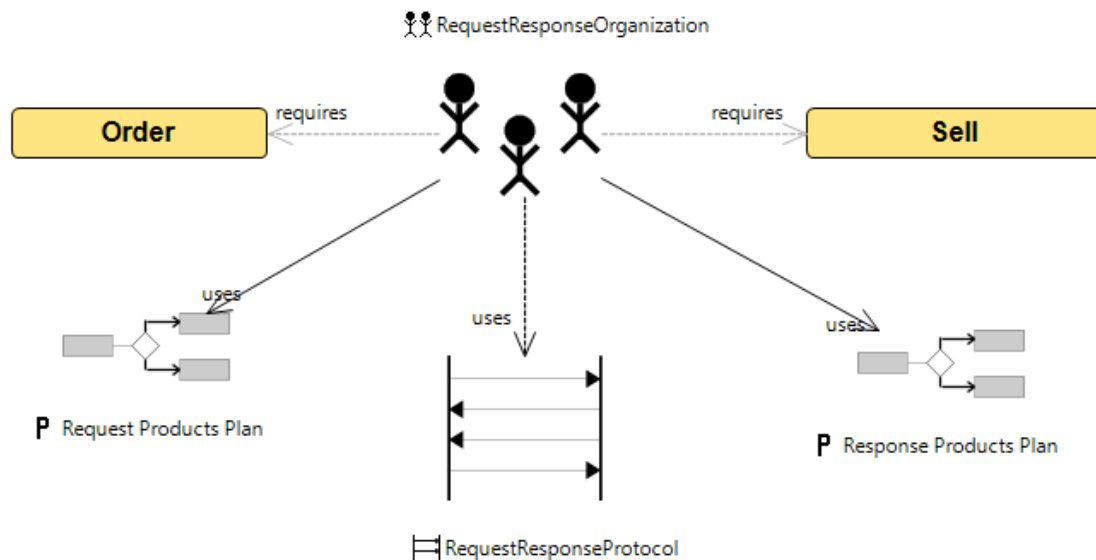


Figura 7.13: [DSML4MAS](#) ponto de vista *Organization*

Os agentes *Request Agent* e *Response Agent* têm uma função, *Order* e *Sell*, respetivamente. Quando o agente *RequestAgent* efetuar uma solicitação para efetuar uma ordem de compra, o agente *ResponseAgent* executa a função de encontrar os produtos *RequestProductsPlan*.

Por outro lado, o agente *ResponseAgent*, pode executar a função *ResponseProductsPlan* para vender os seus produtos. Ambos os planos realizam a sua descoberta interagindo com o agente *RequestResponseMatcher* que regista os serviços aplicando o plano *RequestResponseRegister*. O agente *RequestAgent* e *ResponseAgent* aplicam o plano *Agreement*, para verificar que os dois agentes estão de acordo, sendo que essa interação é feita através das interfaces *RequestProducts* e *ResponseProducts* do SWS *RequestResponseService*. Por último, caso o resultado da interação for positivo, os agentes aplicam o plano *StockOrder* que irá chamar o *Web Request Response* executando o processo *SendRequestProcess* e *SendOrderProcess* usando o *RequestResponseAccess*.

A Fig. 7.14, mostra uma solução possível na linguagem `SEA_ML++` enquanto a Fig. 7.15, mostra uma solução possível na linguagem `DSML4MAS`.

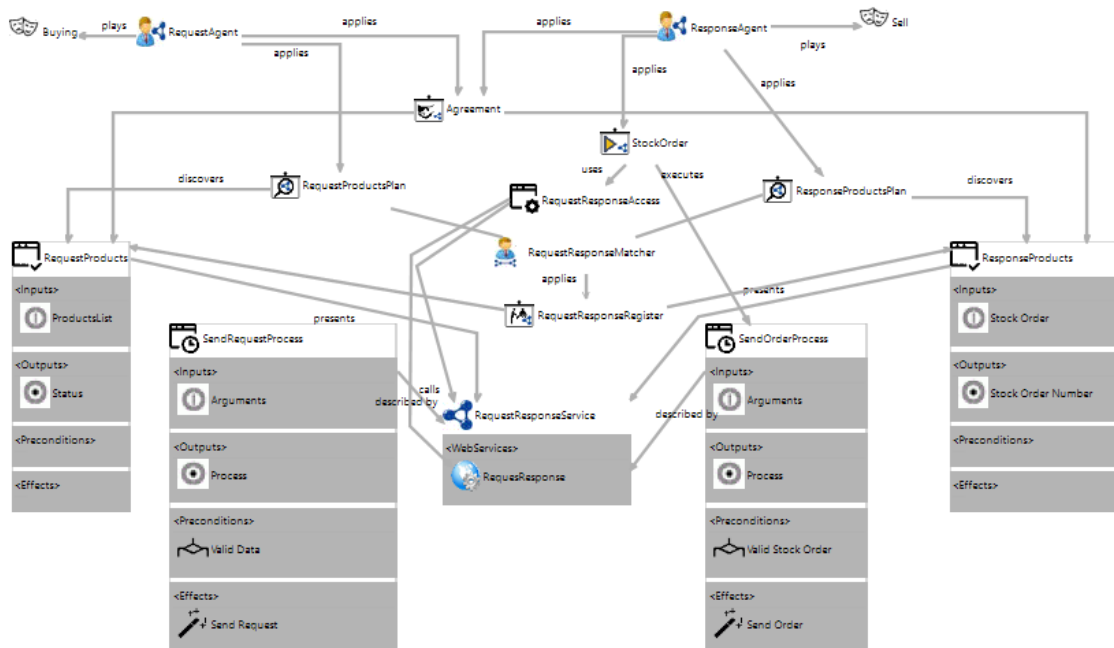
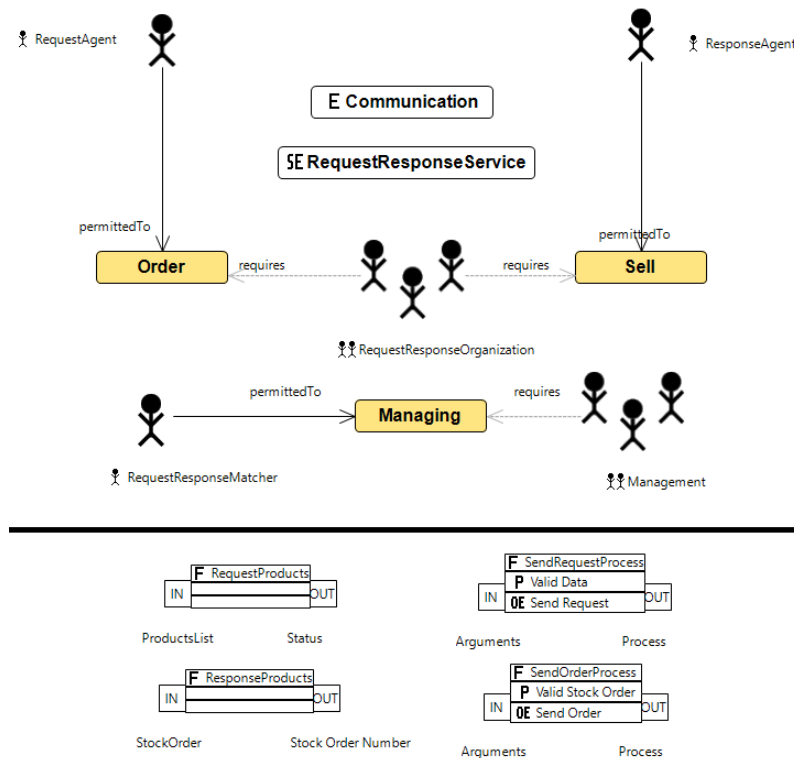


Figura 7.14: **SEA_ML++** ponto de vista *SWSInteraction*

Figura 7.15: [DSML4MAS](#) ponto de vista *Mas*

7.6 Plano de Execução

O plano de execução (Fig. 7.16) está dividido nos seguintes passos:

1. Cada utilizador irá individualmente sentar-se em frente ao computador;
2. É feita uma apresentação de ambas as linguagens que o utilizador irá realizar para desenvolver o caso de teste;
3. É fornecida uma carta de consentimento para que o utilizador a possa ler e assinar;
4. O *Eye Tracker* será calibrado para uso individual para verificar que o sistema está a funcionar corretamente. Nesta fase, apenas foram aceites calibrações cujo resultado fosse entre Excelente e Bom. Caso contrário, era pedido ao utilizador que efetuasse uma nova calibração no sistema. Esta calibração é feita com a ajuda da plataforma *Ogama* [Ogaa], onde são também introduzidos alguns dados sobre o participante como o género, idade, sexo e alguns comentários que sejam necessários;
5. No monitor onde o utilizador irá realizar o processo experimental é possível observar a área onde o utilizador irá resolver o exercício e o enunciado com o exercício a resolver;
6. É registado o momento em que o utilizador começou o exercício;

7. Durante a experiência, para além dos dados registados pelo *Eye Tracker* são também registados comentários efetuados pelo participante e o número de tarefas realizadas corretamente e incorretamente;
8. É registado o momento de término do exercício por parte do utilizador;
9. É fornecido um questionário *SUS* ao participante, para que seja possível contrastar as suas opiniões sobre a usabilidade o *SEA_ML++* e do *DSML4MAS*.

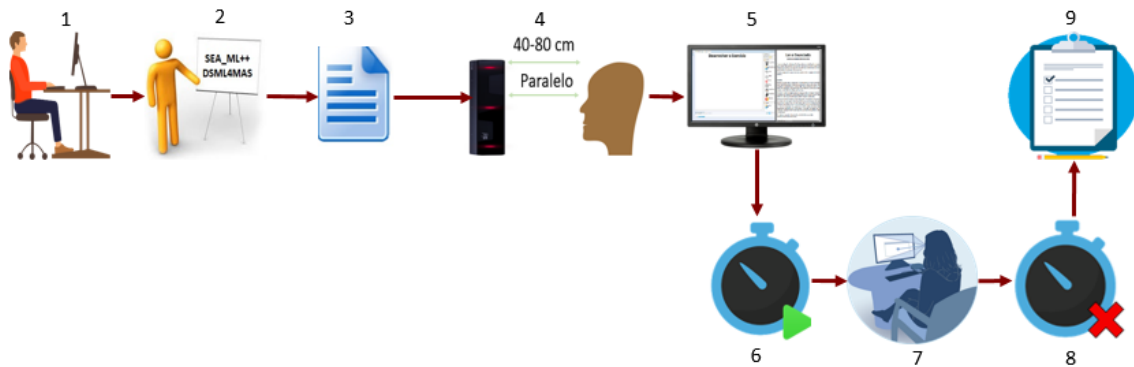


Figura 7.16: Plano de Execução

Com o intuito de reduzir efeitos de aprendizagem, foi trocada a ordem da linguagem com que os participantes começaram a resolver o exercício. 12 participantes começaram o processo experimental utilizando o *SEA_ML++* e 12 participantes começaram o processo experimental utilizando o *DSML4MAS*.

Não foi estabelecido um limite de tempo para a realização deste plano experimental, mas os participantes demoraram em média 20 minutos.

7.7 Preparação dos Dados

Para que fosse possível ser feita uma análise dos dados, foi utilizado a plataforma *Ogama* que para além de permitir guardar todos os dados obtidos pelo *Eye Tracker* através da funcionalidade *Desktop Recording* e guardar dados sobre a interação do participante com o rato, permite também uma análise desses mesmos resultados.

7.7.1 Ogama

Esta plataforma possui vários módulos de análise [Voß09], pelo que nesta experiência foram utilizados:

- *Recording Module* - Neste módulo é possível criar a apresentação que irá guardar os dados do *Eye Tracker* e a tela onde o participante irá resolver os exercícios;
- *Replay Module* - Permite reproduzir os dados registados durante a fase experimental. Muitas configurações podem ser feitas relativamente a visualização dos dados, como

desenhar círculos que representam a posição e o tempo de fixação e desenhar linhas retas que conectam os círculos de fixação;

- *Statistic Module* - Este módulo permite calcular muitos parâmetros úteis como, dados sobre as fixações, as sacadas e do rato. Estes parâmetros podem ser importantes para efetuar uma análise mais detalhada sobre cada participante;
- *Areas of Interest Module* - Permite definir e exibir as áreas de interesse (AOI).

7.7.2 Heatmap

Infelizmente, não é possível construir um *HeatMap* com recurso ao *Ogama* pois como é referido em [Ogab], a análise de *Desktop Recording* não é a sua especialidade, existindo alguns problemas ainda sem resolução. Assim sendo, à medida que o utilizador realizava a experiência, para além do *Ogama* foi necessário executar um novo software [Sof], que para além de guardar um vídeo com todos os passos realizados pelo participante na experiência, guarda também um ficheiro *.csv* com todas as posições que o utilizador visualizou e o respetivo tempo em milissegundos. Foram feitas algumas melhorias neste programa, no sentido em que sempre que o participante não se encontrava a olhar para o ecrã, aparece uma mensagem no vídeo indicando que o participante poderia estar a escrever, pelo que o seu olhar se encontrava fixado no teclado.

De seguida, para a obtenção de um *HeatMap* num vídeo foram necessários alguns passos:

1. Executar um programa que recebe como input todos os pontos (x,y,t), em que x e y representam as coordenadas e t representa o instante em milissegundos em que essa fixação ocorreu. Como output é gerado um conjunto de imagens (transparentes), exemplo Fig. 7.17, em 20 frames por segundo.

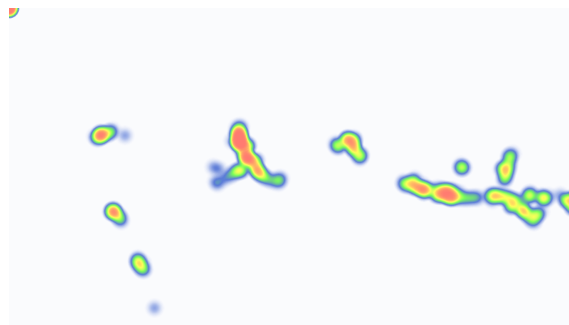


Figura 7.17: Exemplo Heatmap sem Imagem de Fundo

2. Com recurso à biblioteca FFMPEG, foi executado o seguinte comando na linha de comandos: `ffmpeg -i #1 -vf fps=20 #2`, onde -i indica que irá receber um input, #1 é o input com o caminho para o vídeo, -vf indica que serão utilizados filtros neste caso `fps=20`, ou seja, irão ser produzidos 20 frames em 1 segundo e #2 é o caminho para

uma pasta onde vão ser guardadas as imagens. Ao executar este comando, serão reproduzidas todas frames do vídeo, exemplo Fig. 7.18.

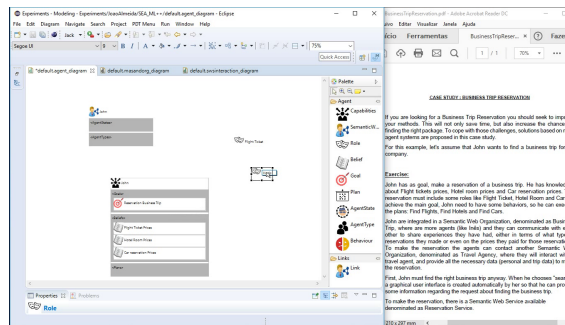


Figura 7.18: Exemplo de Imagem de Fundo

3. Executar um programa em *Python* que recebe as duas imagens dos dois passos anteriores e que irá sobrepor a imagem do passo 1 à imagem do passo 2. Desta forma é possível obter a seguinte imagem exemplo Fig. 7.19.

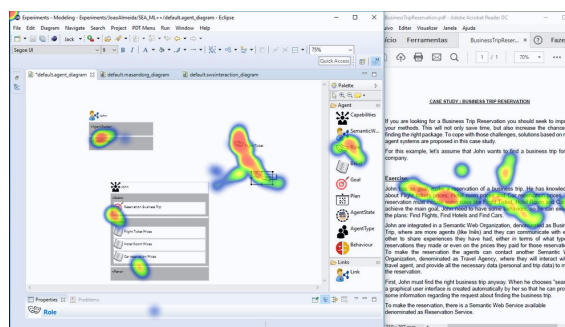


Figura 7.19: Exemplo Heatmap

4. De novo com recurso à biblioteca FFMPEG, foi executado o seguinte comando na linha de comandos: **ffmpeg -framerate 20 -i #1%d.png #2** que irá gerar o vídeo final. Nos parâmetros, 20 indica o número de frames por segundo, #1 indica o caminho para a pasta com as imagens finais, %d indica que irá ser substituído pelo número da imagem e #2 é o caminho onde irá ser gravado o vídeo final com o conjunto de imagens produzidas no passo 3.

Este processo foi moroso, pois os passos 1,3 e 4 foram realizados 6 vezes por cada participante (número de linguagens a testar(2) x número de pontos de vista(3) x numero participantes(24)), totalizando 144 realizações dos diversos passos referidos. O passo 2 foi realizado duas vezes por participante (número de linguagens a testar(2) x vídeo *Desktop Recording*(1) x numero participantes(24)), resultando em 48 vezes.

7.8 Análise dos Dados

Para a análise das 42 sessões criadas pelos 24 participantes (2 por cada participante), foram criadas duas tabelas. Na primeira tabela pretende-se apontar os elementos criados pelos participantes e numa segunda tabela as ligações efetuadas entre esses elementos criados. Em ambas as tabelas o número de linhas corresponde ao número de participantes que realizaram os casos de teste e o número de colunas corresponde aos elementos e ligações que são necessários criar para a realização correta dos exercícios. A tabela foi preenchida com 0 e 1, em que o número 1 indica que o participante criou o elemento ou a ligação corretamente e o número 0 indica que o participante não criou esse elemento ou ligação.

Para a análise destas tabelas, foram utilizadas as seguintes métricas de forma a avaliar os três componentes de eficácia cognitiva (exatidão, velocidade e facilidade).

- *Avaliar a Exatidão:*

- *Precision* - Fração de tarefas realizadas corretamente pelo participante sobre o número total de tarefas realizadas pelo participante;
- *Recall* - Fração de tarefas realizadas pelo participante corretamente sobre o número total de tarefas necessárias para a realização do exercício corretamente;
- *F-measure* - Média ponderada da precisão e recall, conhecido pelo Teste de Precisão;

- *Avaliar a Velocidade:*

- *Duração* - Tempo que o utilizador demora para a realizar a tarefa (expresso em min:seg);

- *Avaliar a Facilidade:*

- *SUS* - Mede a perceção do esforço cognitivo despendido na realização das tarefas;
- *Cliques* - Mede o número de cliques que os participantes fizeram para realizar um determinado exercício;
- *Tarefas Incorretas* - Número médio de tarefas realizadas incorretamente pelo participante.

Com recurso a estas métricas foi possível efetuar uma análise dos [SMA](#) modelados pelos participantes. Na Fig. 7.20 são apresentadas as métricas citadas anteriormente, para a média dos dois exercícios em ambas as linguagens. Adicionalmente, será feita uma análise mais detalhada considerando outras ferramentas estatísticas - Tabela 7.1.

Em relação ao recall, esta apresenta um valor médio de 0,44 e 0,50 para o [SEA_ML++](#) e [DSML4MAS](#) respetivamente, pelo que a diferença entre os valores não é estatisticamente significativa. Ao comparar os 3 pontos de vista, é possível verificar que no [SEA_ML++](#), à

Linguagem	SEA_ML++			DSML4MAS		
ViewPoint	Agent	MasAndOrg	Interaction	Agent	Mas	Organization
Precision	0,83	0,78	0,88	0,86	0,70	0,91
Precision (Média)	0,83			0,82		
Recall	0,56	0,47	0,31	0,45	0,35	0,70
Recall (Média)	0,44			0,50		
F-Measure	0,67	0,58	0,46	0,59	0,46	0,79
F-Measure (Média)	0,57			0,61		
Duração Média	00:05:22	00:02:43	00:03:22	00:04:06	00:02:58	00:01:08
Duração Total Média	00:11:27			00:08:11		
Cliques	177			182		
Tarefas Incorretas	1,04	2,17	0,5	0,75	1,17	0,34
Tarefas Incorretas (Média)	1,24			0,75		
SUS	41,57			33,65		

Figura 7.20: Métricas Eficácia Cognitiva

exceção do ponto de vista *Agent* que possui uma recall de 0,56, os restantes possuem um recall abaixo de 0,50. Um dos fatores que pode influenciar estes valores, principalmente no ponto de vista *Interaction*, é o facto de que para a modelação destes pontos de vista não ser suficiente a leitura do enunciado, isto é, é necessário bons conhecimentos sobre a linguagem, conhecimentos esses que os participantes em questão não possuem. Em relação ao ponto de vista agente, era esperado que os participantes o resolvessem corretamente pois tudo o que era necessário modelar era dito no enunciado. Assim, depois de uma visualização dos "Desktop Recording" dos participantes em ambos os exercícios, foi possível detetar algumas ausências semelhantes por parte dos participantes:

- **Ligação Cooperates** - No [SEA_ML++](#) esta ligação é usada quando dois agentes pretendem interagir. Nos 100% dos participantes que criaram este tipo de ligação, apenas criaram uma ligação do agente A para um outro agente B. Assim, apenas o agente A pode comunicar com o agente B e o inverso não será possível. No exercício 1 é dito que "*they can communicate with each other to share experiences they have had*", pelo que devem existir duas ligações entre os agentes em questão: uma ligação do agente A para o agente B e uma outra ligação do agente B para o agente A (exemplo Fig. 7.21);

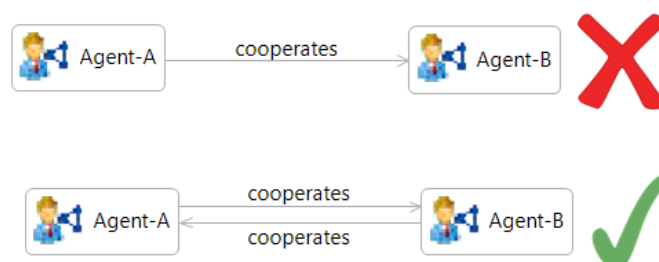


Figura 7.21: Ligação Cooperates

Em relação à precisão, esta apresenta um valor médio aproximadamente igual em ambas as linguagens, aproximadamente 0,83, pelo que novamente a diferença entre os valores não é estatisticamente significativa. Ao comparar a precisão nos 3 pontos de vista para o [SEA_ML++](#), o valor mais baixo é para o ponto de vista MasAndOrg, sendo também possível verificar pelo número de tarefas incorretas que apresenta um valor maior neste ponto de vista (aproximadamente 2). Assim, foi necessário analisar qual a causa deste valor, um pouco mais baixo em comparação com os restantes. Foi possível concluir acerca deste ponto de vista, que existem erros comuns por partes dos participantes:

- **Agentes e Organizações** - No [SEA_ML++](#) os agentes devem interagir com as Organizações através da ligação *works_in*, indicando assim que um determinado agente trabalha para uma determinada organização. Um organização possui ainda um *Compartment* que tem como objetivo criar sub-organizações. No enunciado é dito que: "*John are integrated in a Semantic Web Organization, denominated as Business Trip, where are more agents (like Inês)*", pelo que seria necessário criar duas ligações: uma ligação do tipo *works_in* entre o Agente *John* e a organização *Business Trip* e uma outra ligação do mesmo tipo entre o agente *Inês* e a organização *Business Trip*. Aproximadamente 42% dos participantes, ignoraram a ligação *works_in* e tentaram inserir os dois agentes dentro do *Compartment* da organização *Business Trip* (exemplo Fig. 7.22);

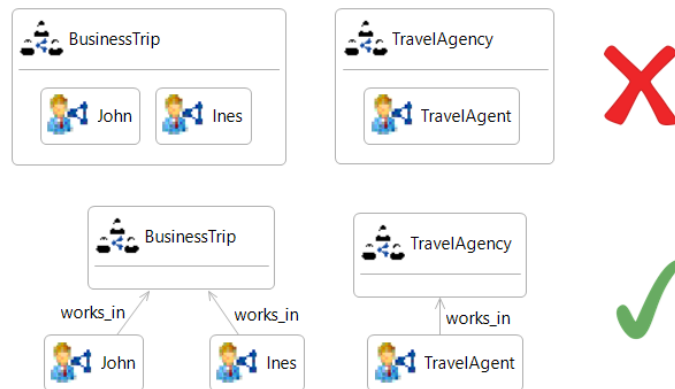


Figura 7.22: Agentes dentro das Organizações

- **Ligação *Interacts_with*** - No [SEA_ML++](#) esta ligação é usada quando as organizações pretendem partilhar recursos, sendo necessário um objeto *Environment* para que seja possível essa troca. No enunciado é dito que: "*To make the reservation the agents can contact another Semantic Web Organization, denominated as Travel Agency*". Como ambos os agentes trabalham na mesma organização, *Business Trip*, deve existir uma ligação do tipo *interacts_with*, entre a organização *Business Trip* e um objeto *Environment*, que pode ser denominado como *Communication*, e uma outra ligação do mesmo tipo entre a organização *Travel Agency* e o mesmo objeto *Environment*.

Assim, ambas as organizações têm acesso ao mesmo ambiente de forma a poderem partilhar todos os recursos necessários (exemplo Fig. 7.23);

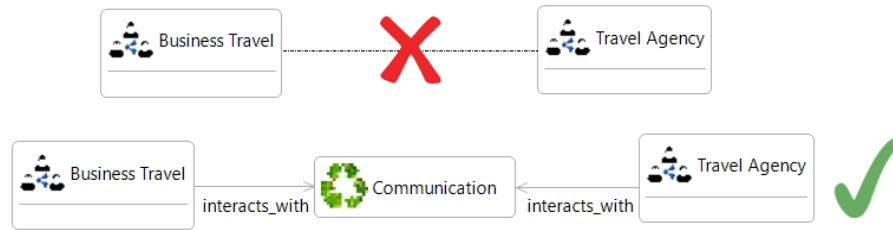


Figura 7.23: Organizações e Ambientes

Sobre o ponto de vista agente, também existem alguns erros comuns por parte dos participantes:

- **Goal e Capability** - No [SEA_ML++](#) para criar um *Goal*, primeiro era necessário que os participantes criassem um objeto *Capability*, pois este possui um *Compartment* onde é possível incluir outros três objetos: *Goal*, *Beliefs* e *Plans*. Cerca de 58% dos participantes, tentaram criar um objeto *Goal* antes de criar um objeto *Capability*.
- **Knowledge e Capability** - No [DSML4MAS](#) em relação ao objeto *Knowledge*, este assume um papel importante no ponto de vista agente, indicando todos os conhecimentos que um determinado agente possui, enquanto o objeto *Capability* pretende transmitir as competências que um agente tem para realizar as suas funções. Para a criação de um objeto *Knowledge*, era necessário clicar sobre o agente em que se pretende definir essas características, onde de seguida o participante era dirigido para um outro ponto de vista interno do agente. Cerca de 50% dos participantes não clicaram sobre o agente. Não encontrando o objeto *knowledge*, acabaram por definir os conhecimentos do agente com recurso ao objeto *Capability*.

Os resultados para a métrica F-Measure revelam um valor médio aproximadamente igual quer para o [SEA_ML++](#) quer para o [DSML4MAS](#), 0,57 e 0,61 respetivamente.

Em relação à métrica duração, esta apresenta uma média de 11min:27seg para o [SEA_ML++](#) e de 08min:11seg para o [DSML4MAS](#). Quando comparados os 3 pontos de vista para ambas as linguagens, é possível verificar que o [SEA_ML++](#) apresenta uma maior duração em todos os pontos de vista. Assim, é possível verificar que os participantes demoram mais tempo a resolver o mesmo exercício com a linguagem [SEA_ML++](#), quando comparada com a [DSML4MAS](#). Apesar disso, quando comparados os resultados do questionário de satisfação (*SUS*), é possível verificar que o [SEA_ML++](#) apresenta melhores valores em comparação com o [DSML4MAS](#), 41,57 e 33,65 respetivamente. De acordo com [\[Ban+08\]](#), a média aceitável do *SUS* situa-se entre os 65-84 pontos. Assim é possível confirmar pela opinião dos participantes que ambas as linguagens estão muito longe de cumprir os requisitos mínimos de efetividade, eficiência e satisfação.

Como já foi referido anteriormente, neste processo experimental foi necessário dividir a tela onde o utilizador realizava o exercício em duas partes, sendo a maior parte ocupada pelo editor e uma parte menor com o caso de estudo que o participante deve modelar. Assim, para uma melhor análise, foi necessário analisar os dados com base em (AOI). Estas áreas correspondem a uma área do estímulo que são definidas de modo a permitir gerar dados estatísticos sobre o comportamento dos participantes nessas mesmas áreas. Deste modo, como é possível observar pela Fig. 7.24, foram definidas 3 AOI:

- **Eclipse Development:** Maior área ocupada pela tela do ecrã, onde o participante modela o [SMA](#);
- **Symbols:** Área onde os participantes irão encontrar os objetos e conexões necessárias para modelar o [SMA](#);
- **PDF:** Área onde o participante pode ler o enunciado com as especificações para modelar o [SMA](#).

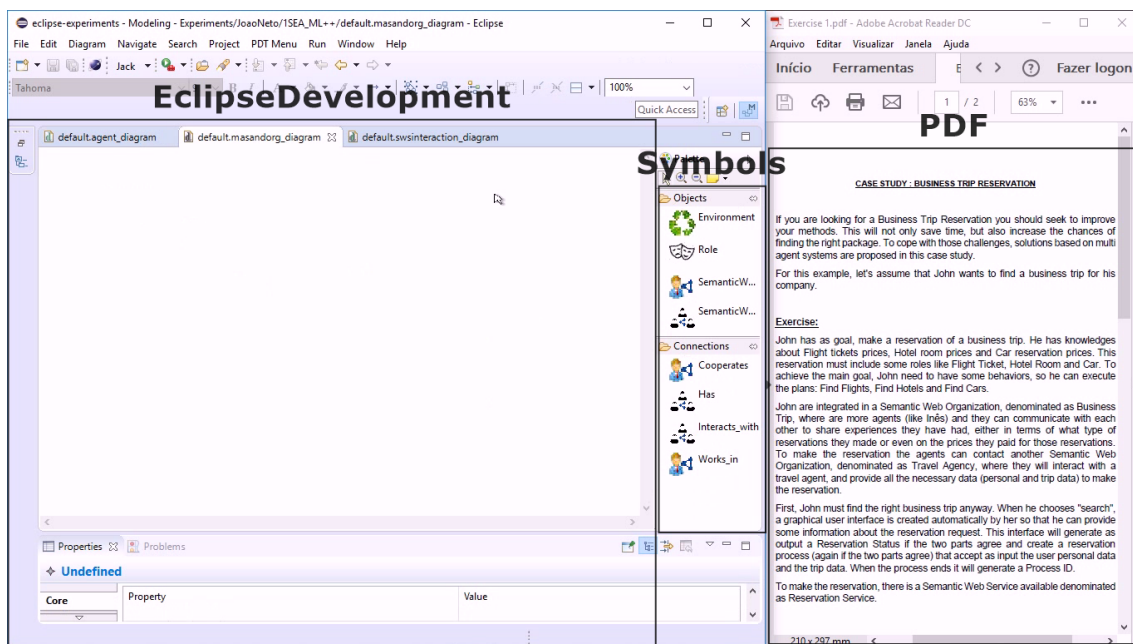


Figura 7.24: Áreas de Interesse

Para a análise destas áreas de interesse, foram utilizadas métricas baseadas em fixações e sacadas, necessárias para perceber qual o esforço cognitivo por parte dos participantes:

- **Número de Fixações** - Número de Fixações está diretamente relacionado com a eficiência. De acordo com [Doh+10], um maior número de fixações pode indicar uma menor eficiência, e no sentido inverso um menor número de fixações pode indicar uma maior eficiência;
- **Tempo de Fixação** - Duração total das fixações numa determinada AOI;

- **Duração de Fixação (Média)** - A Duração Média de Fixação, pode indicar que valores mais altos são geralmente considerados como uma maior dificuldade por parte do participante para compreender as informações;
- **Número de Sacadas** - Número total de movimentos entre fixações ocorridas com destino a uma determinadas AOI;
- **Duração de Sacadas (Média)** - Média de duração de sacada numa determinadas área de interesse;
- **Tamanho de Sacadas (Média)** - Comprimento médio de uma sacada;
- **Regressão** - Número de sacadas regressivas, isto é, número de sacadas que ocorrem com um movimento à esquerda. Ocorre quando um participante volta a visualizar um determinada posição logo após ter mudado de fixação. Apenas foi avaliada esta métrica para a *AOI-Symbols*, pois como os símbolos se encontram à esquerda e a sua descrição à direita, pretende-se perceber se existiu alguma dificuldade por parte do participante em compreender os símbolos.

Para a análise destas métricas foi criada uma tabela (Fig. 7.25), onde o número de linhas corresponde às métricas que se pretende analisar e o número de colunas corresponde às AOI para as quais se pretende avaliar essas métricas. O número de colunas foi duplicado, de forma a poder validar as métricas para ambas as linguagens, *SEA_ML++* e *DSML4MAS*.

	SEA_ML++			DSML4MAS		
	Eclipse Development	Symbols	PDF	Eclipse Development	Symbols	PDF
Número de Fixações	492	206	452	393	108	292
Tempo de Fixação	213456	80941	118476	167150	50763	79160
Duração de Fixação - Média	453	407	264	422	413	282
Número de Sacadas	492	205	448	397	107	289
Duração Sacada - Média	155	118	116	155	98	136
Tamanho Sacada - Média	107	67	100	132	51	103
Regressão	*	17	*	*	9	*

Figura 7.25: Métricas Eye Tracker

Em relação ao número de fixações, estas apresentam valores superiores em todas as AOI no *SEA_ML++*. Assim, podemos afirmar que existiu por parte dos participantes um maior esforço visual para explorar, interpretar e navegar nestas AOI, quando comparado com o *DSML4MAS*. O tempo de fixação também é superior em todas as AOI no *SEA_ML++*. Um dos fatores que pode influenciar este valor quando comparados com o *DSML4MAS*, é o facto da duração do desenvolvimento do SMA com o *SEA_ML++* ser superior. A duração média de fixação do *SEA_ML++* também apresenta valores superiores, ainda assim a diferença entre os valores em ambas as linguagens não é significativa.

Relativamente ao número de sacadas, tal como o número de fixações, também apresentam um maior valor em todas as AOI no *SEA_ML++*. Desta forma podemos concluir que existe um maior esforço visual, significando que um participante realiza uma pesquisa

pouco eficiente. Em relação à duração e tamanho médio de sacada, estes apresentam valores idênticos em ambas linguagens.

A regressão para a *AOI-Symbols*, da mesma forma apresenta um valor superior no [SEA_ML++](#) podendo assim indicar alguma dificuldade por parte dos participantes na compreensão dos símbolos. Apesar disso, quando analisado os *Desktop Recording* para ambas as linguagens, é possível verificar que a grande maioria dos participantes apenas olha para as descrições, ignorando assim os símbolos gráficos (exemplo Fig. 7.26). Assim, estas regressões podem apenas significar uma releitura das descrições.

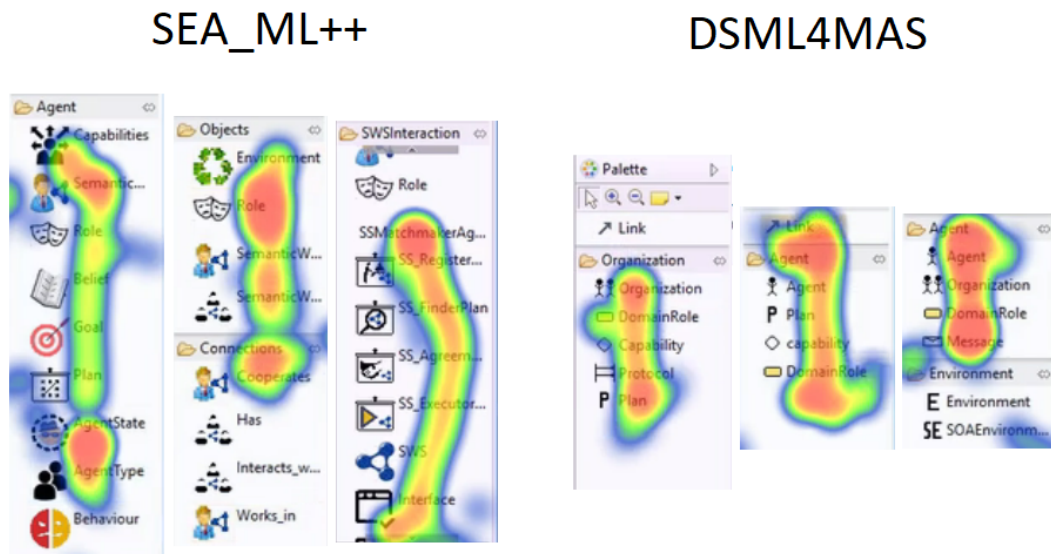


Figura 7.26: HeatMap *AOI-Symbols*

Foram também observados os *HeatMaps* obtidos para cada participante e como se pode observar pela Fig. 7.27, é possível verificar que a área com uma maior carga cognitiva, quer para o [SEA_ML++](#) quer para o [DSML4MAS](#), é a *AOI-Symbols*. Além disso, outro aspeto bastante relevante é o facto de existir uma maior carga cognitiva na *AOI-PDF* quando é utilizada o [SEA_ML++](#) para modelar um *SMA*. Isto pode evidenciar que os participantes se sentem perdidos quando utilizam o [SEA_ML++](#), necessitando de ler o enunciado várias vezes de forma a perceber que objetos serão precisos para a modelação do *SMA*.

Foram também visualizadas as sequências de ações que os participantes realizavam para a modelação do *SMA* em ambas as linguagens. Na figura 7.28, é possível verificar dois exemplos de participantes que realizaram o processo experimental. É possível aferir que para o ponto de vista *Agent* ambos criaram praticamente os mesmos elementos mas em tempos distintos. Na figura 7.28(a), o participante renomeou o objeto *Capability* para *John* e apagou o objeto *Behaviour* que tinha sido criado anteriormente. Nas figuras 7.28(c) e 7.28(d), encontram-se as ligações efetuadas e os respetivos tempos em que foram feitas. Deste modo, é possível concluir que ambos os participantes optaram por efetuar as ligações no fim do exercício e com ordens muito semelhantes.

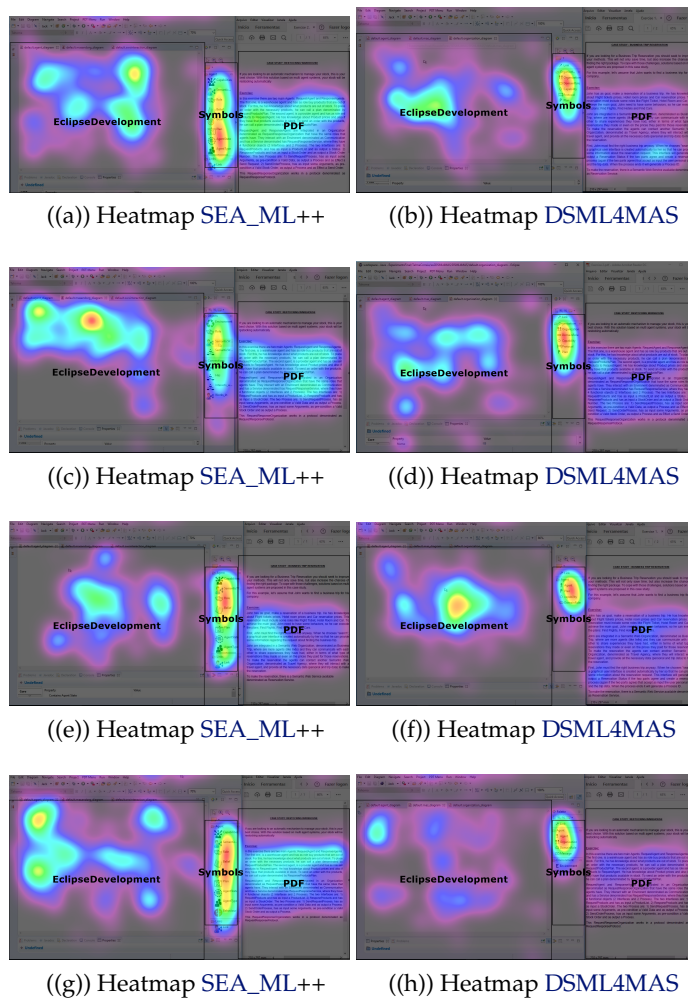


Figura 7.27: HeatMaps

Foram criados e analisados cerca de 12 processos e uma vez que os participantes leem os mesmos enunciados, seria previsível que realizassem aproximadamente a mesma sequência de ações, em tempos diferentes.

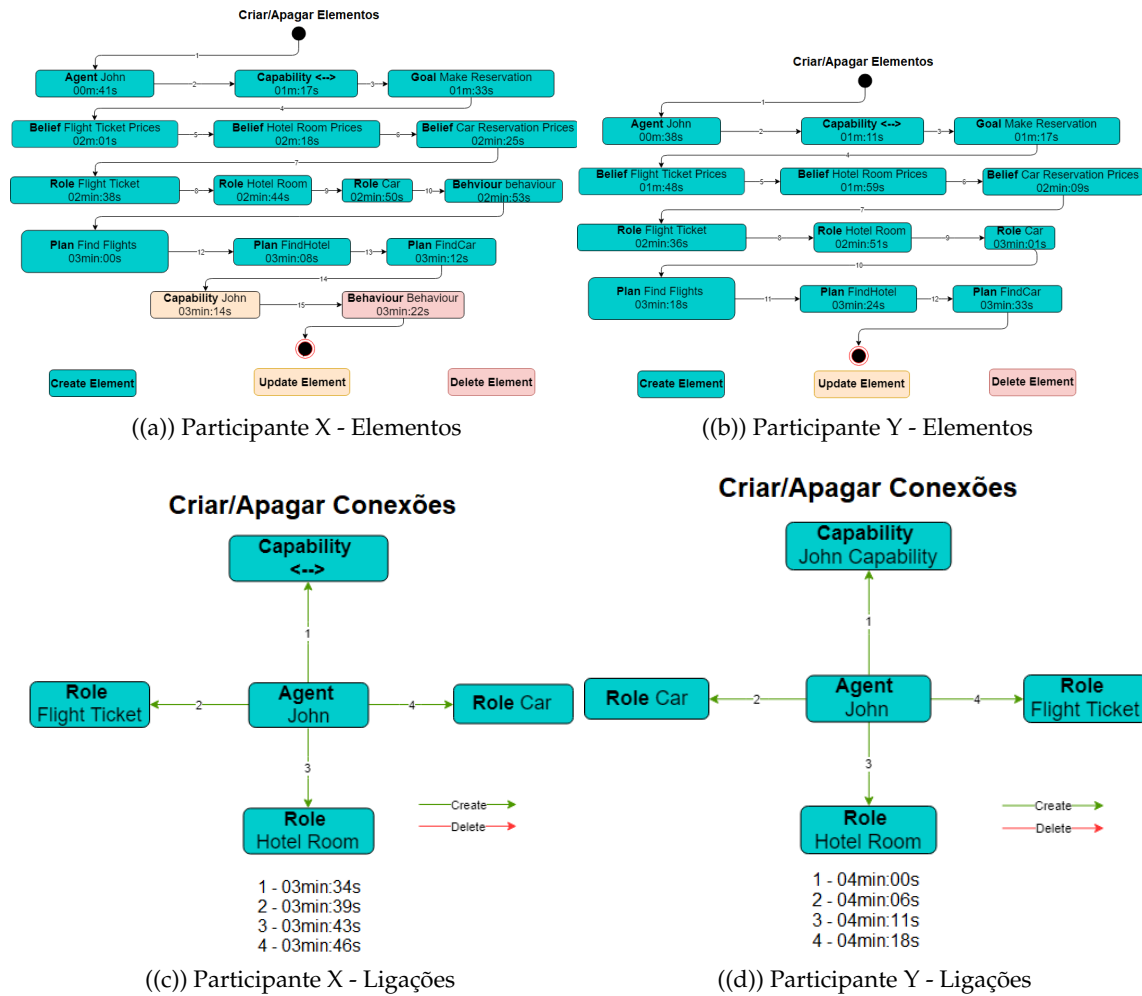


Figura 7.28: Elementos e Ligações

7.8.1 Estatística Descritiva

Depois de feita uma análise geral, foram recolhidas várias medidas de estatística descritiva para analisar os dados e obter uma visão geral da sua distribuição. Deste modo foi analisada a média, a mediana, o desvio padrão, a assimetria, a curtose e o Teste de Shapiro Wilk. A tabela 7.1 sumaria estes resultados, sendo que para cada métrica existem 4 linhas: duas linhas para a linguagem `SEA_ML++` para ambos os exercícios e duas para a linguagem `DSML4MAS` também para ambos os exercícios. Nas colunas são apresentadas as medidas de estatística descritiva. Ao longo das respostas às perguntas de investigação, serão da mesma forma apresentados alguns *boxplots* com a distribuição das métricas que contrastam os resultados para ambas as linguagens.

Métrica	Linguagem	Exercício	Média	Mediana	Desvio Padrão	Assimetria	Curtose	Shapiro-Wilk
Recall	SEA_ML++	1	0,31	0,29	0,17	0,06	-1,16	0,062
		2	0,58	0,63	0,15	0,73	2,08	0,002
	DSML4MAS	1	0,42	0,41	0,14	0,25	-1,13	0,008
		2	0,60	0,57	0,28	0,10	-1,43	0,001
Precision	SEA_ML++	1	0,82	0,82	0,10	0,06	-0,83	0,202
		2	0,83	0,85	0,07	-0,56	-0,61	0,018
	DSML4MAS	1	0,84	0,87	0,15	-0,54	-0,96	0,002
		2	0,83	0,80	0,11	0,19	-0,96	0,005
F-Measure	SEA_ML++	1	0,42	0,42	0,19	-0,26	-0,98	0,082
		2	0,68	0,70	0,11	-0,31	1,39	0,090
	DSML4MAS	1	0,55	0,56	0,15	0,02	-1,03	0,080
		2	0,65	0,68	0,26	-0,30	-0,55	0,021
Cliques	SEA_ML++	1	190	142	111	1,27	0,00	0,003
		2	191	174	54	1,06	0,35	0,123
	DSML4MAS	1	82	169	51	-0,06	-0,73	0,719
		2	86	164	55	0,99	0,43	0,168
Duração	SEA_ML++	1	689	489	443	1,44	0,66	0,002
		2	684	609	217	1,39	1,32	0,036
	DSML4MAS	1	468	436	219	0,72	0,43	0,805
		2	599	510	246	1,27	1,34	0,058
SUS	SEA_ML++	*	42	43	14	-0,17	-0,66	0,700
	DSML4MAS	*	34	34	16	0,40	0,50	0,582
Número de Fixações	SEA_ML++	1	401	261	296	1,65	1,34	0,001
		2	365	357	73	0,63	0,29	0,751
	DSML4MAS	1	254	235	118	1,90	4,87	0,013
		2	275	244	89	0,48	-1,47	0,103
Tempo de Fixação	SEA_ML++	1	141144	99270	91797	1,56	1,22	0,002
		2	134105	136060	25549	-0,44	-0,41	0,887
	DSML4MAS	1	92564	86791	36575	1,07	2,21	0,272
		2	105485	90351	36120	0,69	-1,12	0,067
Duração Média de Fixação	SEA_ML++	1	374	388	36	-1,16	1,65	0,194
		2	376	381	29	-0,34	1,43	0,777
	DSML4MAS	1	383	384	31	-0,51	-0,18	0,691
		2	411	407	45	1,22	2,82	0,151
Número de Sacadas	SEA_ML++	1	401	261	296	1,65	1,34	0,001
		2	361	353	73	0,77	0,62	0,526
	DSML4MAS	1	253	235	118	1,89	4,84	0,013
		2	275	244	89	0,47	-1,47	0,104
Tamanho Sacada	SEA_ML++	1	95	97	6	0,27	-0,94	0,410
		2	88	87	5	1,41	2,56	0,102
	DSML4MAS	1	98	98	8	0,47	0,03	0,805
		2	93	93	5	0,58	0,29	0,887
Duração Média de Sacada	SEA_ML++	1	139	134	25	-0,17	-0,34	0,558
		2	120	120	29	0,03	0,31	0,961
	DSML4MAS	1	128	114	35	0,05	-1,09	0,241
		2	131	131	25	-0,44	-0,84	0,268

Tabela 7.1: Estatística Descritiva

RQ1: Será que o exercício resolvido usando o [SEA_ML++](#) se encontra mais próximo do objetivo pretendido no caso de teste, em comparação com o [DSML4MAS](#)?

Em relação à média do recall, como é possível verificar pela tabela 7.1 e comprovado pela figura 7.29, possui um maior valor no [DSML4MAS](#) em comparação com o [SEA_ML++](#). Deste modo, é possível concluir que na primeira linguagem os participantes realizam um maior número de tarefas corretamente.

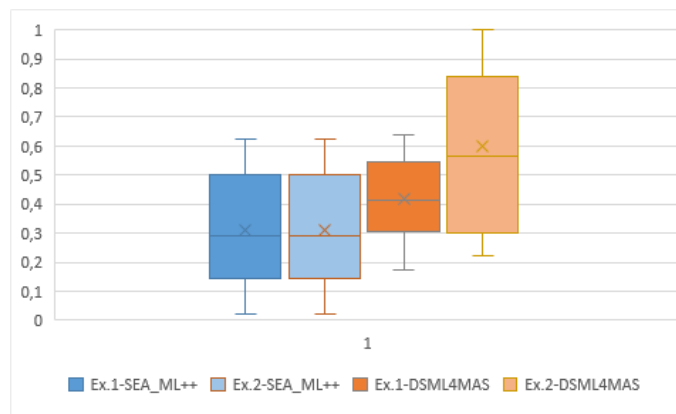


Figura 7.29: Recall para ambas as linguagens

Verificando os valores da assimetria para os dois exercícios, ambas as linguagens apresentam valores superiores a 0 indicando que a cauda mais longa da distribuição encontra-se à direita, havendo assim uma menor ocorrência de valores mais altos. Este indicador, consegue dar-nos a percepção que a maioria dos participantes tem um baixo grau de recall, isto é, apresenta um valor baixo de tarefas cobertas pelo participante.

Em relação à curtose, à exceção do exercício 2 na linguagem [SEA_ML++](#), todos apresentam um valor negativo mostrando que os dados se encontram muito distribuídos, não existindo assim uma grande concentração dos dados. No exercício 2 na linguagem [SEA_ML++](#), o curtose apresenta um valor muito superior a 0 o que significa que houve uma grande concentração de valores altos na distribuição, existindo também valores muito baixos, o que torna a sua média mais baixa.

Quando analisado o valor do teste Shapiro-Wilk, este apresenta um valor superior a 0,05 apenas na linguagem [SEA_ML++](#) para o primeiro exercício. Neste caso, podemos concluir que estamos perante uma distribuição normal, isto é, simétrica em relação à sua média.

	Estatística	df1	df2	Welch's t-test Sig. (p-value)
Recall	2,998	1	137,737	0,086

Tabela 7.2: Welch's Test

Dito isto, podemos concluir que apesar de o [DSML4MAS](#) apresentar, de uma forma geral melhores resultados, é possível verificar pela Tabela 7.2 que esta diferença entre

linguagens não é estatisticamente significativa uma vez que o p-value, 0,086, é superior a 0,05.

RQ2: Será que desenvolver um sistema no [SEA_ML++](#) requer menos esforço em comparação com o [DSML4MAS](#)?

Para avaliar o esforço cognitivo dos participantes em ambas as linguagens, é necessário recorrer a métricas visuais como, o número, tempo e duração média de fixações assim como o número, tamanho e duração médias das sacadas.

Em relação ao número e tempo de fixações, é possível verificar que em ambos os exercícios os participantes apresentam um valor de média e de mediana superiores no [SEA_ML++](#) em comparação com o [DSML4MAS](#), demonstrando a existência de um maior esforço cognitivo por parte dos participantes nas [AOI](#) para modelar um [SMA](#) com o [SEA_ML++](#). Os valores de assimetria, à exceção do tempo de fixações para o exercício 2 no [SEA_ML++](#), apresentam valores superiores a 0 o que indica uma maior frequência de valores baixos de fixações. Os valores da curtose são todos superiores a 0 à exceção da linguagem [DSML4MAS](#) para o exercício 2 que apresenta um valor negativo.

O número de sacadas também apresenta um valor superior para o [SEA_ML++](#) em comparação com o [DSML4MAS](#), existindo uma maior interação entre as [AOI](#) definidas.

Neste contexto podemos também analisar os valores de F-Measure, uma vez que este combina a precisão com o recall. Para o primeiro exercício o [SEA_ML++](#) apresenta piores resultados em comparação com o [DSML4MAS](#), situação que não se constata no exercício 2 onde a diferença entre linguagens é inferior.

	Estatística	df1	df2	Welch's t-test Sig. (p-value)
F-Measure	1,972	1	140,159	0,162

Tabela 7.3: Welch's Test

É possível concluir de forma geral que desenvolver um [SMA](#) com o [SEA_ML++](#) requer um maior esforço cognitivo em comparação com o [DSML4MAS](#). Ainda assim, através da Tabela 7.3, concluímos que essa diferença não é estatisticamente significativa, uma vez que o p-value, 0,162, é superior a 0,05.

RQ4: Será que com o uso do [SEA_ML++](#) o utilizador apresenta uma maior taxa de satisfação em comparação com o [DSML4MAS](#)?

De acordo com [Ten+10], é possível reconhecer os componentes de qualidade de Nielsen nas diversas questões do [SUS](#): Facilidade de aprendizagem perguntas 3,4,7,10; Eficiência perguntas 5,6,9; Facilidade de Memorização pergunta 2; Minimização de Erros pergunta 6; Satisfação perguntas 1,4,9. Ao analisar o gráfico 7.30, é possível verificar que os participantes ficam mais satisfeitos, apresentam uma maior eficiência e entendem que existe uma maior facilidade de aprendizagem com o uso do [SEA_ML++](#). Por outro lado, os participantes consideram que com o uso do [DSML4MAS](#), existe uma maior facilidade de memorização e minimização da taxa de erros quando modelam um [SMA](#).

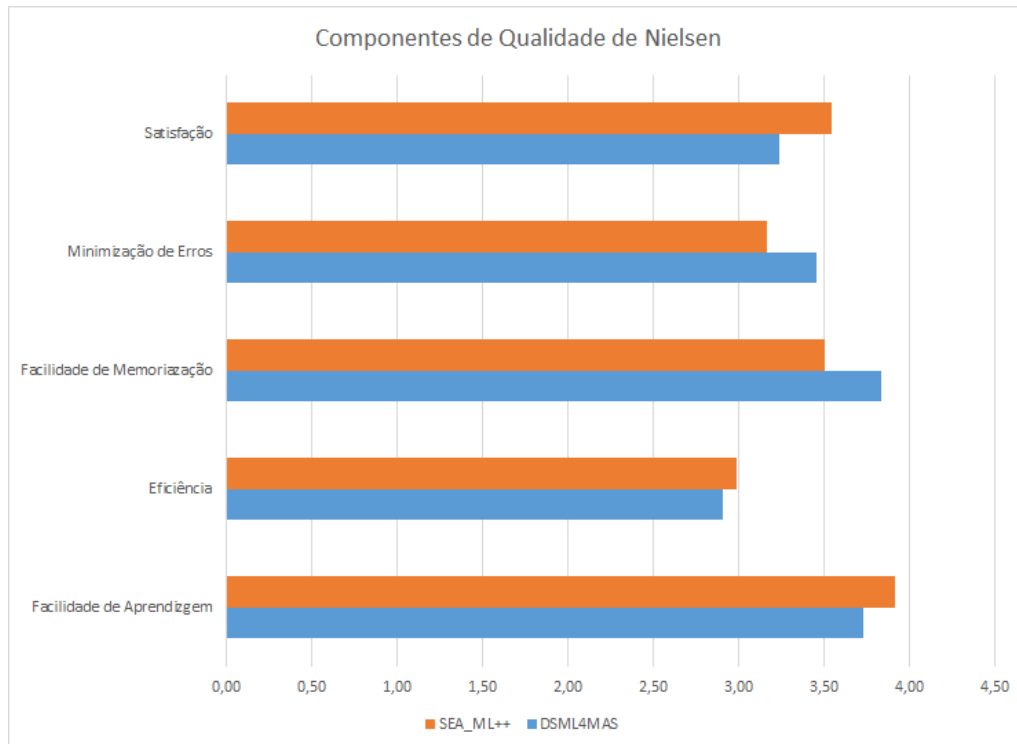


Figura 7.30: Componentes de Qualidade de Nielsen

Para avaliar a taxa de satisfação do participante, recorremos aos resultados do questionário [SUS](#). A hipótese a testar é que o [SEA_ML++](#) apresenta uma melhor classificação quando comparado com o [DSML4MAS](#). De acordo com o teste Welch's apresentado na tabela 7.4, os resultados não têm grande divergência entre si:

$$t_{SEA_ML++}(13,779) = 23; \quad p_{SEA_ML++} = ,000; \quad t_{DSML4MAS}(10,402) = 23; \quad p_{DSML4MAS} = ,000.$$

Welchs Test						
	t	df	p-value	Média	95% Intervalo de Confiança de Diferença	
					Mais Abaixo	Mais Alto
SEA_ML++	13,779	23	,000	41,563	35,323	47,803
DSML4MAS	10,402	23	,000	33,646	26,955	40,337

Tabela 7.4: Welchs Test

Em termos de média, para a linguagem [SEA_ML++](#) os participantes classificaram-na como 42 pontos enquanto que para o [DSML4MAS](#) os participantes deram 33 pontos. O intervalo de confiança de 95% de diferença para o [SEA_ML++](#) encontra-se entre os 35 e 48 pontos enquanto que no [DSML4MAS](#) situa-se entre os 27 e 40 pontos.

De acordo com [LS09], o valor médio para os questionários [SUS](#) é de 70 pontos, indicando uma baixa satisfação dos participantes quer para a linguagem [SEA_ML++](#) quer para a linguagem [DSML4MAS](#). Ainda assim, visto o valor médio ser maior na linguagem [SEA_ML++](#), é possível afirmar, de uma forma geral, que existe um maior grau de satisfação

por parte dos participantes com o uso desta linguagem.

RQ5: Depois de ter sido feito uma apresentação da linguagem, será que o desenvolvimento de sistemas com o `SEA_ML++` é uma tarefa complexa para principiantes em comparação com o `DSML4MAS`?

Para responder a esta pergunta de investigação, serão considerados os participantes principiantes, aqueles que não possuem conhecimentos sobre `LDE`.

Ao analisar a métrica cliques da tabela 7.5, é possível verificar que na linguagem `SEA_ML++` são necessários, em média, o dobro dos cliques em comparação com o `DSML4MAS`. Isto pode significar que existe por parte dos participantes um esforço maior para modelar um `SMA` com o uso da primeira linguagem. Adicionalmente, os participantes também demoram mais tempo para modelar um `SMA` com o uso do `SEA_ML++`, com um tempo médio de 686 segundos que corresponde a 11min:26seg, ao invés do `DSML4MAS` que demoram cerca de 533 segundos que corresponde a 8min:53seg.

Métrica	Linguagem	Exercício	Média	Mediana	Desvio Padrão	Assimetria	Curtose	Shapiro-Wilk
Clique	SEA_ML++	1	156	134	90	2,76	8,21	0,000
		2	197	178	52	1,11	0,25	0,055
	DSML4MAS	1	75	168	53	-0,01	-0,98	0,711
		2	66	144	43	1,03	0,60	0,235
Duração	SEA_ML++	1	546	464	314	2,82	8,54	0,000
		2	705	638	215	1,37	1,15	0,043
	DSML4MAS	1	470	424	228	0,68	0,10	0,870
		2	555	453	245	1,98	4,34	0,009

Tabela 7.5: Métrica Cliques e Duração para Participantes Principiantes

Complementado o referido anteriormente, ao facto de existir um maior número de fixações e sacadas no `SEA_ML++` superior ao `DSML4MAS`, é possível afirmar também que desenvolver um `SMA` com o `SEA_ML++` é uma tarefa com um maior nível de complexidade para principiantes em comparação com o `DSML4MAS`.

RQ7: Será que os participantes ao usarem o `SEA_ML++` cometem menos erros em comparação com o `DSML4MAS`?

Ao analisar a tabela 7.1, é possível verificar que os valores da métrica *Precision* e *F-Measure* apresentam valores muito semelhantes em ambas as linguagens. Em relação à *Precision*, como é possível verificar pela Fig. 7.31(a), ambas as linguagens apresentam valores medianos acima dos 0,80, o que indica que a maioria dos participantes realizou um maior número de tarefas corretamente. Se fizermos 1-Precision obtemos 0,2, assim conseguimos apurar o valor de erros cometidos pelos participantes, uma vez que 1 representa a amostra total de tarefas realizadas. Também é possível concluir pelo bloxplot da figura 7.31(a) que na linguagem `DSML4MAS` o exercício 1 apresenta uma maior discrepância de valores, variando estes entre os 0,5 e 1.

Em relação ao *F-Measure*, pela Fig. 7.31(b) é possível verificar que no exercício 2 na linguagem `DSML4MAS` e exercício 1 na linguagem `SEA_ML++`, existe uma maior diversidade de valores.

Ainda assim, pela Tabela 7.6 e Tabela 7.3, podemos verificar que essa diferença não é

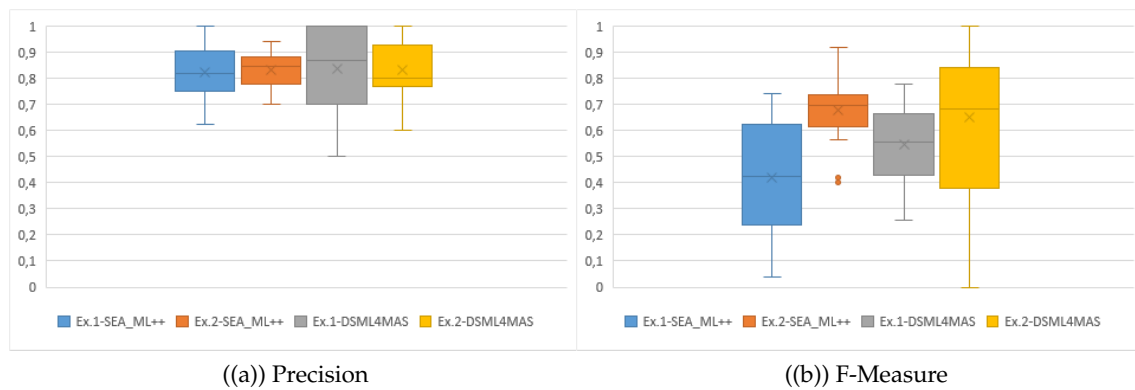


Figura 7.31: Métricas Precision e F-Measure

	Estatística	df1	df2	Welch's t-test Sig. (p-value)
Precision	0,041	1	122,034	0,840

Tabela 7.6: Welch's Test

estatisticamente significativa, uma vez que em ambos os casos o p-value apresenta valores superiores a 0,05, 0,840 e 0,162.

Dito isto, não é possível afirmar de forma genérica que com o uso do [SEA_ML++](#) os participantes cometem menos erros em comparação com o [DSML4MAS](#).

7.9 Ameaças à validade do estudo

Neste estudo, tal como no estudo anterior, foram utilizados os mesmos participantes pelo que as ameaças relativamente ao número de participantes e à forma como foram seleccionados apresentam um risco, pois os resultados obtidos podem não ser iguais quando usada uma amostra de maior dimensão. Além disso, o facto de os utilizadores não conhecerem as linguagens torna uma ameaça à validade deste estudo.

Foi utilizado um *Eye tracker*, especificamente um *Eye Tribe Tracker*. Este dispositivo permite recolher dados sobre o processo cognitivo de alguns utilizadores, sendo possível realizar um mapeamento entre a posição dos olhos do utilizador e o local aproximado que está a observar. Infelizmente, este dispositivo tem uma margem de erro de 0,5 a 1 cm e apenas suporta ecrãs com no máximo 22 polegadas, impondo limitações no seu uso e preparação experimental.

MELHORIAS PROPOSTAS

Depois de desenvolvidas diversas análises e de identificar alguns erros comuns por parte dos participantes, será relevante propor melhorias para ambas as linguagens de forma a aumentar a usabilidade, principalmente do [SEA_ML++](#).

8.0.1 Melhorias [SEA_ML++](#)

Através do estudo empírico anterior verificou-se que 100% dos participantes quando efetua uma ligação do tipo *Cooperates* entre dois agentes, apenas cria uma ligação entre o agente A e o agente B, não tendo em consideração a ligação contrária (entre o agente B e o agente A). Assim seria relevante que sempre que um utilizador tentasse criar uma ligação deste tipo fosse mostrado um pop-up onde seria questionado se ambos os agentes podem comunicar um com o outro. Caso o participante preencha a flag, então irá aparecer uma seta bidirecional entre os dois agentes (Fig. 8.1), caso contrário, iria aparecer uma seta unidirecional entre os dois agentes que o utilizador tentou conectar inicialmente. Deste modo, para além de se tornar possível indicar o tipo de comunicação entre os agentes apenas como uma seta (unidirecional ou bidirecional), conseguimos também reduzir o número de ligações necessárias no diagrama, visto que agora quando dois agentes pretenderem cooperar entre si, apenas será necessário uma ligação.

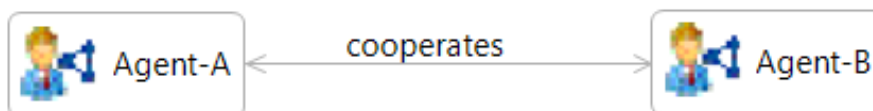


Figura 8.1: Seta Bidirecional entre dois Agentes

Visto que cerca de 42% dos participantes, não utilizaram a ligação *works_in*, consideramos que esta ligação pode ser omitida. Para tal, em vez de um agente estar conectado a

uma organização indicando que esse agente trabalha para uma determinada organização, cada agente é inserido dentro do *Compartment* da organização (Fig. 8.2). Deste modo, é possível não só reduzir o número de ligações na linguagem, como também tornar o ponto de vista *MasAndOrg* mais perceptível, no sentido em que todos os agentes que trabalham para uma determinada organização, encontram-se no interior da mesma.

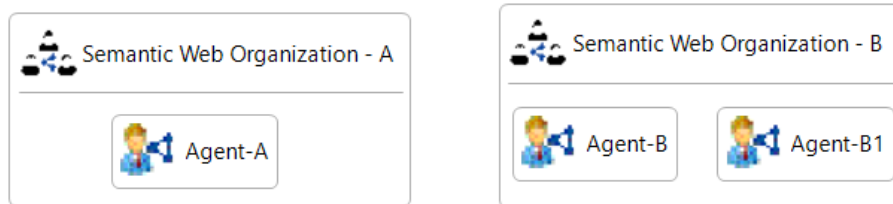


Figura 8.2: Agentes no *Compartment* da Organização

Verificou-se que cerca de 64% dos participantes tiveram dificuldades em definir quer *Goals*, *Beliefs* e *Plans* (8% não definiram mesmo estes conceitos), deste modo considera-se que o objeto *Capability*, onde é possível definir todos estes objetos, torna-se confuso para o participante. Sendo cada agente definido num ponto de vista *Agent* individualmente, não surge o caso de dois agentes partilharem do mesmo objeto *Capability*. Assim, seria benéfico alterar o meta-modelo da linguagem, eliminando o objeto *Capability* e agregar todos estes conceitos no *Compartment* do objeto *Agent* (Fig. 8.3).

optou-se por eliminar este objeto e agregar todos estes conceitos no *Compartment* do objeto *Agent* (Fig. 8.3).

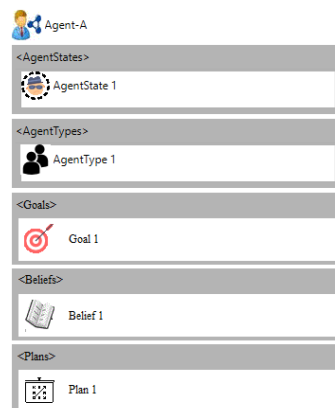


Figura 8.3: *Compartment Agent*

No *SEA_ML++* caso sejam necessários X agentes do tipo A , então é necessário criar X pontos de vista todos iguais, podendo apenas mudar o nome que identifica o agente. Uma solução possível para este problema, é criar um novo ponto de vista denominado de *Deployment*, onde possa ser possível criar instâncias dos agentes previamente criados. Deste modo, apenas seria necessário criar um agente do tipo A e de seguida no ponto de vista *Deployment* seriam criados X instâncias do agente A (Fig. 8.4).

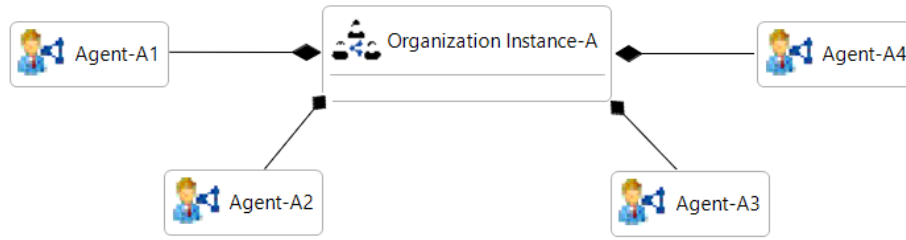


Figura 8.4: *Deployment Viewpoint*

8.0.2 Melhorias DSML4MAS

Ao longo do processo experimental, 100% dos participantes tentaram criar uma ligação entre um agente e uma organização e não tiveram sucesso, pois o meta-modelo da linguagem não o permite. Como faz sentido poder indicar que um agente pertence a uma determinada organização, tal como no [SEA_ML++](#), seria preferível alterar o meta-modelo da linguagem de forma a permitir que essa ligação pudesse ser efetuada (Fig. 8.5).



Figura 8.5: Ligação entre Agentes e Organização

Ao longo do processo experimental, era pedido aos participantes que definissem as organizações e um objeto *Environment* para partilha de recursos entre elas. A totalidade dos participantes que criaram um objeto *Environment*, tentaram mas não conseguiram efetuar uma ligação com as organizações, pois ao contrário do [SEA_ML++](#), o meta-modelo do [DSML4MAS](#) não permite. Assim, seria vantajoso alterar o meta-modelo da linguagem de modo a permitir que essa ligação pudesse ser efetuada. Desse modo, seria possível indicar várias organizações e estabelecer a forma como estas cooperam entre si (Fig. 8.6).

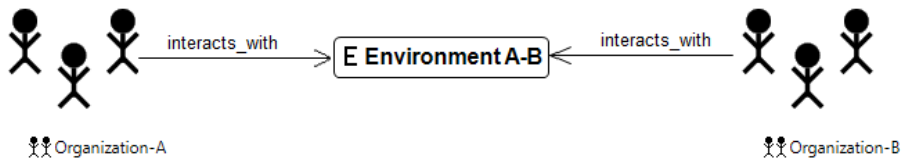


Figura 8.6: Ligação entre Organizações e *Environment*

Um dos conceitos que não está definido no [DSML4MAS](#), ao contrário do [SEA_ML++](#), é a possível definição de sub-organizações. Através deste conceito, é possível referir as diferentes organizações que se podem encontrar dentro de uma organização. Para a definição destas sub-organizações, seria benéfico replicar o método usado no [SEA_ML++](#) sobre como estas sub-organizações podem ser definidas. É definido um *Compartment*

no objeto organização e sempre que um utilizador quiser inserir uma sub-organização, apenas tem de arrastar o objeto organização para dentro do *Compartment* (Fig. 8.7).

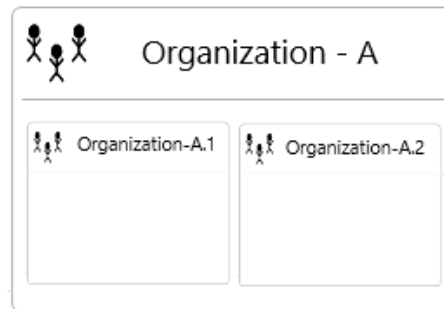


Figura 8.7: *Compartment* para Organizações

No decorrer do processo, quando os participantes pretendem definir conceitos de um determinado agente, cerca de 55% dos participantes criam *Capability* ao invés de *Knowledges*. Num dos enunciados é dito que "*he has knowledge about what products are out of stock*", pelo que o participante deve definir um objeto *knowledge* denominado como "*produtcs our of stock*". O participante quando abre o ponto de vista *Agent*, não encontra o objeto *Knowledge* pois este só está acessível quando o utilizador clica duas vezes sobre o agente ao qual pretende definir o *Knowledge*. Assim, quando o participante abre o ponto de vista *Agent* apenas encontra o objeto *Capability* pelo que acaba por definir um *Knowledge* como uma *Capability*. Deste modo e visto que quando se clica duas vezes sobre um determinado agente apenas é possível definir *Knowledges*, seria vantajoso alterar o meta-modelo da linguagem, transferindo este conceito para o ponto de vista principal *Agent* (Fig. 8.8).

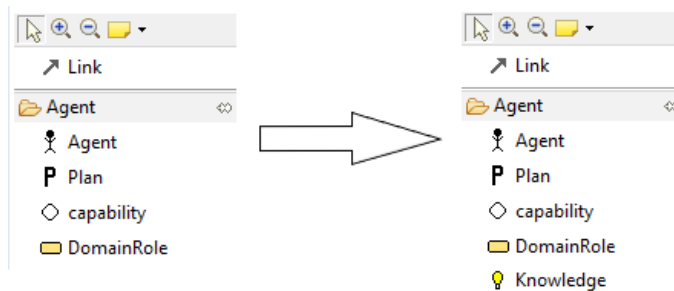


Figura 8.8: DSML4MAS *Agent Viewpoint*

CONCLUSÕES

Numa primeira fase, este trabalho teve como objetivo principal avaliar um dos fatores que ficou por avaliar na primeira iteração feita ao [SEA_ML](#), a transparência semântica. Para tal foi replicado o estudo efetuado por [\[Cai+13\]](#), em ambas as linguagens ([SEA_ML](#) e [SEA_ML++](#)). Numa segunda fase, foi feita uma comparação do [SEA_ML++](#) com uma outra LDE para [SMA](#), [DSML4MAS](#), de maneira a ser possível detetar problemas de usabilidade que possam estar presentes no [SEA_ML++](#). Para a realização destas experiências foi utilizado um conjunto de 24 participantes, sendo que cada participante realizou as duas experiências. Por fim, e através da análise dos resultados obtidos no estudo empírico, foram propostas algumas melhorias tanto para o [SEA_ML++](#) como para o [DSML4MAS](#).

Em relação à transparência semântica, dos 32 símbolos analisados foi verificado que houve uma melhoria de cerca de 12% na taxa de sucesso, que se situa nos 53,8%. No que diz respeito ao coeficiente de transparência semântica a sua melhoria rondou os 13%, situando-se assim este indicador nos 33%. Ainda assim, os resultados sugerem que apesar da sintaxe concreta do [SEA_ML++](#), proposta por [\[Mir17\]](#), apresentar melhores resultados em comparação com o [SEA_ML](#), existe ainda necessidade de desenvolver a compreensibilidade de alguns símbolos.

Relativamente aos resultados obtidos na comparação entre o [SEA_ML++](#) e o [DSML4MAS](#), foi possível responder a algumas perguntas de investigação que haviam sido feitas antes de iniciar o processo experimental. Foi possível verificar que os participantes apresentam uma maior satisfação ao desenvolver um [SMA](#) com o [SEA_ML++](#). Ainda assim, essa satisfação não se reflete no esforço cognitivo apresentado pelos participantes, onde estes revelam um maior esforço cognitivo e uma maior complexidade com o uso desta linguagem.

Através da análise dos resultados obtidos por parte dos participantes, foram propostas algumas melhorias para ambas as linguagens. Apesar do objetivo principal passar por

melhorar o [SEA_ML++](#) do ponto de vista da sua usabilidade, tornou-se relevante expandir esta análise ao [DSML4MAS](#).

Nesta dissertação houve algumas ameaças à validade dos estudos, nomeadamente o número de participantes que foi utilizado em ambas as avaliações e o facto da inexistência de utilizadores experientes. Ainda assim, juntamente com a universidade de *Ege*, irá ser desenvolvida uma replicação deste estudo que contará com utilizadores experientes neste tipo de linguagens.

9.1 Contribuições

Com recurso a um conjunto de métricas para avaliar a eficácia cognitiva, foi possível verificar que podiam ser feitas algumas melhorias na linguagem [SEA_ML++](#). Assim, foi proposto um conjunto de eixos de progressão na linguagem. Adicionalmente, foram também propostas algumas melhorias à linguagem [DSML4MAS](#) de forma a tornar a linguagem mais perceptível para os utilizadores.

Uma parte do trabalho realizado nesta dissertação, nomeadamente o segundo estudo - Avaliar o [SEA_ML++](#) comparando-a com o [DSML4MAS](#) - foi publicado na *Third International Workshop on Human Factors in Modeling (HuFaMo'18)* [JSK18].

9.2 Trabalho Futuro

Como trabalho futuro, pretende-se alargar a escala do processo experimental com mais participantes, e principalmente, com participantes experientes em [SMA](#).

No segundo processo experimental, os participantes realizavam o exercício no [SEA_ML++](#) em 3 pontos de vista: *Agent*, *MasAndOrg* e *Interaction*. Seria também interessante testar os restantes pontos de vista, de forma a verificar onde existe espaço para melhoria.

Seria da mesma forma relevante realizar novamente todo o processo experimental com as melhorias propostas já desenvolvidas. Desta forma seria possível verificar se os eixos de progressão sugeridos resultam em melhorias significativas.

BIBLIOGRAFIA

- [Ame09] D. Ameller. “Considering non-functional requirements in model-driven engineering”. Tese de mestrado. Universitat Politècnica de Catalunya, 2009.
- [Art+09] A. Artikis, G. Picard e L. Vercoouter. *Engineering Societies in the Agents World IX: 9th International Workshop, ESAW 2008, Saint-Etienne, France, September 24-26, 2008, Revised Selected Papers*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009. ISBN: 9783642025624. URL: <https://books.google.pt/books?id=uY1rCQAAQBAJ>.
- [Ban+08] A Bangor, P Kortum e J. Miller. “The system usability scale (SUS): An empirical evaluation”. Em: *International Journal of Human-Computer Interaction* 24.6 (2008), pp. 574–594.
- [Bar+18] A. Barišić, V. Amaral e M. Goulão. “Usability driven DSL development with USE-ME”. Em: *Computer Languages, Systems & Structures* 51.Supplement C (2018), pp. 118 –157. ISSN: 1477-8424. DOI: <https://doi.org/10.1016/j.cl.2017.06.005>. URL: <http://www.sciencedirect.com/science/article/pii/S1477842417300477>.
- [Bel+01] F. Bellifemine, A. Poggi e G. Rimassa. “Developing Multi-agent Systems with JADE”. Em: *Intelligent Agents VII Agent Theories Architectures and Languages*. Ed. por C. Castelfranchi e Y. Lespérance. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 89–103. ISBN: 978-3-540-44631-6.
- [BL+01] T. Berners-Lee, J. Hendler, O. Lassila et al. “The semantic web”. Em: *Scientific american* 284.5 (2001), pp. 28–37.
- [Bev+15] N. Bevan, J. Carter e S. Harker. “ISO 9241-11 revised: What have we learnt about usability since 1998?” Em: *International Conference on Human-Computer Interaction*. Springer. 2015, pp. 143–151.
- [Bor+06] R. H. Bordini, L. Braubach, M. Dastani, A. E. F. Seghrouchni, J. J. Gomez-Sanz, J. Leite, G. O’Hare, A. Pokahr e A. Ricci. “A survey of programming languages and platforms for multi-agent systems”. Em: *Informatica* 30.1 (2006).
- [Bor+09] S. Borsci, S. Federici e M. Lauriola. “On the dimensionality of the System Usability Scale: a test of alternative measurement models”. Em: *Cognitive Processing* 10.3 (2009), pp. 193–197. ISSN: 1612-4790. DOI: [10.1007/s10339-](https://doi.org/10.1007/s10339-009-0000-0)

- 009-0268-9. URL: <https://doi.org/10.1007/s10339-009-0268-9>.
- [Bro+96] J. Brooke et al. "SUS-A quick and dirty usability scale". Em: *Usability evaluation in industry* 189.194 (1996), pp. 4–7.
- [Cai+13] P. Caire, N. Genon, P. Heymans e D. L. Moody. "Visual notation design 2.0: Towards user comprehensible requirements engineering notations". Em: *2013 21st IEEE International Requirements Engineering Conference (RE)*. 2013, pp. 115–124. DOI: [10.1109/RE.2013.6636711](https://doi.org/10.1109/RE.2013.6636711).
- [CR94] V. Caldiera e H. D. Rombach. "The goal question metric approach". Em: *Encyclopedia of software engineering* 2.1994 (1994), pp. 528–532.
- [Cao+09] L. Cao, B. Ramesh e M. Rossi. "Are Domain-Specific Models Easier to Maintain Than UML Models?" Em: *IEEE Software* 26.4 (2009), pp. 19–21. ISSN: 0740-7459. DOI: [10.1109/MS.2009.87](https://doi.org/10.1109/MS.2009.87).
- [Cha+14] M. Challenger, S. Demirkol, S. Getir, M. Mernik, G. Kardas e T. Kosar. "On the use of a domain-specific modeling language in the development of multiagent systems". Em: *Engineering Applications of Artificial Intelligence* 28.Supplement C (2014), pp. 111–141. ISSN: 0952-1976. DOI: <https://doi.org/10.1016/j.engappai.2013.11.012>. URL: <http://www.sciencedirect.com/science/article/pii/S0952197613002297>.
- [Cha+16] M. Challenger, G. Kardas e B. Tekinerdogan. "A systematic approach to evaluating domain-specific modeling language environments for multi-agent systems". Em: *Software Quality Journal* 24.3 (2016), pp. 755–795. ISSN: 1573-1367. DOI: [10.1007/s11219-015-9291-5](https://doi.org/10.1007/s11219-015-9291-5). URL: <https://doi.org/10.1007/s11219-015-9291-5>.
- [CL05] R. Choren e C. Lucena. "Modeling Multi-agent systems with ANote". Em: *Software & Systems Modeling* 4.2 (2005), pp. 199–208. ISSN: 1619-1374. DOI: [10.1007/s10270-004-0065-y](https://doi.org/10.1007/s10270-004-0065-y). URL: <https://doi.org/10.1007/s10270-004-0065-y>.
- [Cos+03] M.-F. Costabile, D. Fogli, C. Letondal, P. Mussio e A. Piccinno. "Domain-expert users and their needs of software development". Em: *HCI 2003 End User Development Session*. 2003.
- [DS+08] V. T. Da Silva, R. Choren e C. J. De Lucena. "MAS-ML: a multiagent system modelling language". Em: *International Journal of Agent-Oriented Software Engineering* 2.4 (2008), pp. 382–421.
- [Dix09] A. Dix. "Human-computer interaction". Em: *Encyclopedia of database systems*. Springer, 2009, pp. 1327–1331.
- [Doh+10] S. Doherty, S. O'Brien e M. Carl. "Eye tracking as an MT evaluation technique". Em: *Machine translation* 24.1 (2010), pp. 1–13.

- [DR99] J. S. Dumas e J. Redish. *A practical guide to usability testing*. Intellect books, 1999.
- [Dus+13] F. Duseau, B. Dufour e H. Sahraoui. “Vasco: A Visual Churn Exploration Tool”. Em: *Joint Proceedings of Tools, Demos & Posters* (2013), p. 17.
- [Fer99] J. Ferber. *Multi-agent systems: an introduction to distributed artificial intelligence*. Vol. 1. Addison-Wesley Reading, 1999.
- [Fow10] M. Fowler. *Domain-specific languages*. Pearson Education, 2010.
- [Gur99] C. A. Gurr. “Effective diagrammatic communication: Syntactic, semantic and pragmatic issues”. Em: *Journal of Visual Languages & Computing* 10.4 (1999), pp. 317–342.
- [Hah08] C. Hahn. “A domain specific modeling language for multiagent systems”. Em: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems–Volume 1*. International Foundation for Autonomous Agents e Multiagent Systems. 2008, pp. 233–240.
- [Hut+14] J. Hutchinson, J. Whittle e M. Rouncefield. “Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure”. Em: *Science of Computer Programming* 89.Part B (2014). Special issue on Success Stories in Model Driven Engineering, pp. 144 –161. ISSN: 0167-6423. DOI: <https://doi.org/10.1016/j.scico.2013.03.017>. URL: <http://www.sciencedirect.com/science/article/pii/S0167642313000786>.
- [Jac90] R. J. K. Jacob. “What You Look at is What You Get: Eye Movement-based Interaction Techniques”. Em: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’90. Seattle, Washington, USA: ACM, 1990, pp. 11–18. ISBN: 0-201-50932-6. DOI: 10.1145/97243.97246. URL: <http://doi.acm.org/10.1145/97243.97246>.
- [JSK18] V. A.M.G.B.T.T.O.F.A.M. C. João Silva Ankica Barišić e G. Kardas. “Comparing the Usability of two Multi-Agents Systems DSLs: SEA_ML++ and DSML4MAS StudyDes”. Em: *Third International Workshop on Human Factors in Modeling (HuFaMo’18)*. 2018.
- [Kle08] A. Kleppe. *Software language engineering: creating domain-specific languages using metamodels*. Pearson Education, 2008.
- [LS09] J. R. Lewis e J. Sauro. “The factor structure of the system usability scale”. Em: *International conference on human centered design*. Springer. 2009, pp. 94–103.
- [Mer+05a] M. Mernik, J. Heering e A. M. Sloane. “When and how to develop domain-specific languages”. Em: *ACM computing surveys (CSUR)* 37.4 (2005), pp. 316–344.

- [Mer+05b] M. Mernik, J. Heering e A. M. Sloane. “When and How to Develop Domain-specific Languages”. Em: *ACM Comput. Surv.* 37.4 (dez. de 2005), pp. 316–344. ISSN: 0360-0300. DOI: [10.1145/1118890.1118892](https://doi.org/10.1145/1118890.1118892). URL: <http://doi.acm.org/10.1145/1118890.1118892>.
- [Mir17] T. R. B. de Miranda. “Software Language Engineering: Interaction and Usability Modeling of Languages Editors”. Tese de mestrado. Faculdade de Ciências e Tecnologia da Universidade de Lisboa, 2017.
- [Moo09] D. Moody. “The “physics” of notations: toward a scientific basis for constructing visual notations in software engineering”. Em: *IEEE Transactions on Software Engineering* 35.6 (2009), pp. 756–779.
- [Moo+10] D. L. Moody, P. Heymans e R. Matulevičius. “Visual syntax does matter: improving the cognitive effectiveness of the i* visual notation”. Em: *Requirements Engineering* 15.2 (2010), pp. 141–175.
- [Ngu+14] V.-C. Nguyen, X. Qafmolla e K. Richta. “Domain specific language approach on model-driven development of web services”. Em: *Acta Polytechnica Hungarica* 11.8 (2014), pp. 121–138.
- [Nie03] J. Nielsen. *Usability 101: Introduction to usability*. 2003.
- [Nun07] I. O. de Nunes. “Implementação do Modelo e da Arquitetura BDI”. Em: *Monografias em Ciência da Computação* 1 (2007).
- [Ogaa] *Open Gaze and Mouse Analyser*. URL: <http://www.ogama.net/> (acedido em 02/07/2018).
- [Ogab] *Open Gaze and Mouse Analyser - Changes*. URL: <http://www.ogama.net/node/18> (acedido em 02/07/2018).
- [Pic+11] P. Pichler, B. Weber, S. Zugal, J. Pinggera, J. Mendling e H. A. Reijers. “Imperative versus declarative process modeling languages: An empirical investigation”. Em: *International Conference on Business Process Management*. Springer. 2011, pp. 383–394.
- [RW+11] N. M. Razali, Y. B. Wah et al. “Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests”. Em: *Journal of statistical modeling and analytics* 2.1 (2011), pp. 21–33.
- [SP11] C. Salinesi e O. Pastor, eds. *Advanced Information Systems Engineering Workshops - CAiSE 2011 International Workshops, London, UK, June 20-24, 2011. Proceedings*. Vol. 83. Lecture Notes in Business Information Processing. Springer, 2011. ISBN: 978-3-642-22055-5. DOI: [10.1007/978-3-642-22056-2](https://doi.org/10.1007/978-3-642-22056-2). URL: <https://doi.org/10.1007/978-3-642-22056-2>.
- [Sof] *Saving of gaze data taken by EyeTribe tracker*. URL: <http://miguelgoulao.github.io/EyeT/> (acedido em 02/07/2018).

- [Sch01] M. Schumacher. *Objective Coordination in Multi-agent System Engineering: Design and Implementation*. Berlin, Heidelberg: Springer-Verlag, 2001. ISBN: 3-540-41982-9.
- [SJ11] T. ShaoPeng e Z. Jun. "A research on multi agent modeling language". Em: *Procedia Engineering* 15 (2011), pp. 1842–1847.
- [Sha+15] Z. Sharafi, Z. Soh e Y.-G. Guéhéneuc. "A systematic literature review on the usage of eye-tracking in software engineering". Em: *Information and Software Technology* 67 (2015), pp. 79–107. ISSN: 0950-5849. DOI: <https://doi.org/10.1016/j.infsof.2015.06.008>.
- [Spi01] D. Spinellis. "Notable design patterns for domain-specific languages". Em: *Journal of Systems and Software* 56.1 (2001), pp. 91–99. ISSN: 0164-1212. DOI: [https://doi.org/10.1016/S0164-1212\(00\)00089-3](https://doi.org/10.1016/S0164-1212(00)00089-3). URL: <http://www.sciencedirect.com/science/article/pii/S0164121200000893>.
- [Spr10] A. Sprogis. "The configurator in DSL tool building". Em: *Computer Science and Information Technologies, Scientific Papers, University of Latvia* 756 (2010), pp. 173–192.
- [Ste+08] D. Steinberg, F. Budinsky, E. Merks e M. Paternostro. *EMF: eclipse modeling framework*. Pearson Education, 2008.
- [Ten+10] J. M. Tenório, F. M. Cohrs, V. L. Sdepanian, I. T. Pisa e H. de Fátima Marin. "Desenvolvimento e avaliação de um protocolo eletrônico para atendimento e monitoramento do paciente com doença celíaca". Em: *Revista de Informática Teórica e Aplicada* 17.2 (2010), pp. 210–220.
- [VDK02] A. Van Deursen e P. Klint. "Domain-specific language design requires feature descriptions". Em: *CIT. Journal of computing and information technology* 10.1 (2002), pp. 1–17.
- [VD+00] A. Van Deursen, P. Klint e J. Visser. "Domain-specific languages: An annotated bibliography". Em: *ACM Sigplan Notices* 35.6 (2000), pp. 26–36.
- [Viy+14] V. Viyović, M. Maksimović e B. Perisić. "Sirius: A rapid development of DSM graphical editor". Em: *IEEE 18th International Conference on Intelligent Engineering Systems INES 2014*. 2014, pp. 233–238. DOI: [10.1109/INES.2014.6909375](https://doi.org/10.1109/INES.2014.6909375).
- [Völ+13] M. Völter, T. Stahl, J. Bettin, A. Haase e S. Helsen. *Model-driven software development: technology, engineering, management*. John Wiley & Sons, 2013.
- [Voß09] A. Voßkühler. "OGAMA description (for Version 2.5)". Em: *Berlin, Germany: Freie Universit* (2009).

- [WW06] Y. Wang e Y. Wang. “Cognitive informatics models of the brain”. Em: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 36.2 (2006), pp. 203–207. ISSN: 1094-6977. DOI: [10.1109/TSMCC.2006.871151](https://doi.org/10.1109/TSMCC.2006.871151).
- [Wei99] G. Weiss. *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press, 1999.
- [Yus+07] S. Yusuf, H. Kagdi e J. I. Maletic. “Assessing the Comprehension of UML Class Diagrams via Eye Tracking”. Em: *15th IEEE International Conference on Program Comprehension (ICPC '07)*. 2007, pp. 113–122. DOI: [10.1109/ICPC.2007.10](https://doi.org/10.1109/ICPC.2007.10).



QUESTIONÁRIO DEMOGRÁFICO

Questionário Demográfico utilizado no processo experimental.

Demographic Questionnaire

1. Age

2. Gender

Marcar apenas uma oval.

☐ Male

☐ Female

3. Nationality

4. Are you a student ?

Marcar apenas uma oval.

☐ Yes

☐ No

5. If yes, what is your course?

6. If you have a job, what is your function?

7. Experience with Domain Specific Language (DSL)?

Marcar apenas uma oval.

☐ I've learned DSL in the context of a course.

☐ I've used DSL in a professional context

☐ I know what DSL is but never used it

☐ I've never heard of it.

8. Experience with Multi-Agent Systems (MAS)

Marcar apenas uma oval.

☐ I've learned MAS in the context of a course.

☐ I've used MAS in a professional context

☐ I know what MAS is but never used it

☐ I've never heard of it.

9. Experience with Domain Specific Languages for Multi-Agent Systems (MAS)

Marcar apenas uma oval.

☐ I've learned in the context of a course.

☐ I've used in a professional context

☐ I know what is but never used it

☐ I've never heard of it.

CARTA DE CONSENTIMENTO

Carta de Consentimento utilizada no processo experimental.

Consent Information Letter

Information to participants:

This experimental work is conducted within the NOVA Laboratory for Computer Science and Informatics (NOVA LINCS). NOVA LINCS is a new unit of the national Science & Technology network in the area of Computer Science and Engineering, launched in 2014/2015, and hosted at the Departamento de Informática of Faculdade de Ciências e Tecnologia of Universidade NOVA de Lisboa (DI-NOVA), a leading academic department in Portugal.

All information stated as part of this experiment is confidential and will be kept as such.

Prof. Miguel Goulão and Prof. Vasco Amaral are responsible for this experiment and can be contacted at:

Miguel Goulão: mgoul@fct.unl.pt; +351 21 294 85 36 (ext. 10731); Office: P2/17.

Vasco Amaral: vma@fct.unl.pt, +351 21 294 85 36 (ext. 10712); Office: P2/3.

We would like to emphasize that:

- ✓ your participation is entirely voluntary;
- ✓ you are free to refuse to answer any question;
- ✓ you are free to withdraw at any time.

The experiment will be kept strictly confidential and will be made available only to members of the research team of the study or, in case external quality assessment takes place, to assessors under the same confidentiality conditions. Data collected in this experiment may be part of a final research report, but under no circumstances will your name or any identifying characteristic be included in the report.



QUESTIONÁRIO DE SATISFAÇÃO

Questionário de Satisfação utilizada no processo experimental.

Satisfaction Questionnaire

1. I think that I would like to use this system frequently.

Mark only one oval.

[illegible]

2. I found the system unnecessarily complex.

Mark only one oval.

[illegible]

3. I thought the system was easy to use.

Mark only one oval.

[illegible]

4. I think that I would need the support of a technical person to be able to use this system.

Mark only one oval.

[illegible]

5. I found the various functions in this system were well integrated.

Mark only one oval.

[illegible]

6. I thought there was too much inconsistency in this system.
Mark only one oval.

[illegible]

7. I would imagine that most people would learn to use this system very quickly.
Mark only one oval.

[illegible]

8. I found the system very cumbersome to use.
Mark only one oval.

[illegible]

9. I felt very confident using the system.
Mark only one oval.

[illegible]

10. I needed to learn a lot of things before I could get going with this system.
Mark only one oval.

[illegible]



EXERCÍCIO 1

Exercício 1 denominado por *Business Trip Reservation* utilizada no processo experimental.

CASE STUDY : BUSINESS TRIP RESERVATION

If you are looking for a Business Trip Reservation you should seek to improve your methods. This will not only save time, but also increase the chances of finding the right package. To cope with those challenges, solutions based on multi agent systems are proposed in this case study.

For this example, let's assume that John wants to find a business trip for his company.

Exercise:

John has as goal, make a reservation of a business trip. He has knowledges about Flight tickets prices, Hotel room prices and Car reservation prices. This reservation must include some roles like Flight Ticket, Hotel Room and Car. To achieve the main goal, John need to have some behaviors, so he can execute the plans: Find Flights, Find Hotels and Find Cars.

John are integrated in a Semantic Web Organization, denominated as Business Trip, where are more agents (like Inês) and they can communicate with each other to share experiences they have had, either in terms of what type of reservations they made or even on the prices they paid for those reservations. To make the reservation the agents can contact another Semantic Web Organization, denominated as Travel Agency, where they will interact with a travel agent, and provide all the necessary data (personal and trip data) to make the reservation.

First, John must find the right business trip anyway. When he chooses "search", a graphical user interface is created automatically by her so that he can provide some information about the reservation request. This interface will generate as output a Reservation Status if the two parts agree and create a reservation process (again if the two parts agree) that accept as input the user personal data and the trip data. When the process ends it will generate a Process ID.

To make the reservation, there is a Semantic Web Service available denominated as Reservation Service.



EXERCÍCIO 2

Exercício 2 denominado por *Restocking Warehouse* utilizada no processo experimental.

CASE STUDY: RESTOCKING WAREHOUSE

If you are looking to an automatic mechanism to manage your stock, this is your best choice. With this solution based on multi agent systems, your stock will be restocking automatically

Exercise:

In this exercise there are two main Agents: RequestAgent and ResponseAgent. The first one, is a warehouse agent and has as role buy products that are out of stock. For this, he has knowledge about what products are out of stock. To place an order with the necessary products, he can call a plan denominated by RequestProductsPlan. The second agent, is a provider agent and has as role sell products to RequestAgent. He has knowledge about Product prices and also if they have that products available in stock. To send an order with the products, he can call a plan denominated by ResponseProductsPlan.

RequestAgent and ResponseAgent are integrated in an Organization denominated as RequestResponseOrganization that have the same roles that agents have. They interact with an Environment denominated as Communication and has a Service denominated as RequestResponseService, where they have 4 functional objects (2 Interfaces and 2 Process). The two Interfaces are: 1) RequestProducts and has as input a ProductList. 2) ResponseProducts and has as input a StockOrder. The two Process are: 1) SendRequestProcess, has as input some Arguments, as pre-condition a Valid Data and as output a Process. 2) SendOrderProcess, has as input some Arguments, as pre-condition a Valid Stock Order and as output a Process.

This RequestResponseOrganization works in a protocol denominated as RequestResponseProtocol.



